21世纪高等学校规划教材 | 计算机应用



Web前端设计

——HTML+CSS+jQuery技术教程

吴 强 编著

清华大学出版社

21 世纪高等学校规划教材·计算机应用

Web 前端设计——HTML＋CSS＋jQuery 技术教程

吴　强　编著

清华大学出版社
北　京

内 容 简 介

本书对 Web 设计基础知识与常用工具进行了较全面的讲解，内容包括 HTML、CSS、jQuery 及网页设计的常用工具如 Photoshop、Dreamweaver、Flash。本书针对较新的软件版本 HTML5 及 CSS3 进行介绍，并配有教学课件以帮助进行教学，且提供所有例子的源代码文件。特别地，HTML 和 CSS 为中英文双语内容，有利于 Web 设计双语课的开展。

本书可作为大学本科、专科有关专业的网页设计、Web 应用等课程的教材，也可供广大网站设计用户参考。

图书在版编目(CIP)数据

Web 前端设计：HTML+CSS+jQuery 技术教程/吴强编著. --北京：清华大学出版社，2014
21 世纪高等学校规划教材·计算机应用
ISBN 978-7-302-38358-1

Ⅰ. ①W…　Ⅱ. ①吴…　Ⅲ. ①网页制作工具－高等学校－教材　Ⅳ. ①TP393.092

中国版本图书馆 CIP 数据核字(2014)第 243369 号

责任编辑：闫红梅　薛　阳
封面设计：
责任校对：焦丽丽
责任印制：

出版发行：清华大学出版社
　　网　　址：http://www.tup.com.cn，http://www.wqbook.com
　　地　　址：北京清华大学学研大厦 A 座　　**邮　　编：**100084
　　社 总 机：010-62770175　　**邮　　购：**010-62786544
　　投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
　　质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
　　课件下载：http://www.tup.com.cn，010-62795954
印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：185mm×260mm　**印　张：**20.75　　**字　　数：**501 千字
版　　次：2014 年 12 月第 1 版　　**印　　次：**2014 年 12 月第1 次印刷
印　　数：1～　000
定　　价：　.00 元

产品编号：060356-01

出版说明

随着我国改革开放的进一步深化，高等教育也得到了快速发展，各地高校紧密结合地方经济建设发展需要，科学运用市场调节机制，加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度，通过教育改革合理调整和配置了教育资源，优化了传统学科专业，积极为地方经济建设输送人才，为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是，高等教育质量还需要进一步提高以适应经济社会发展的需要，不少高校的专业设置和结构不尽合理，教师队伍整体素质亟待提高，人才培养模式、教学内容和方法需要进一步转变，学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007 年 1 月，教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》，计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”，通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容，进一步深化高等学校教学改革，提高人才培养的能力和水平，更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中，各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势，对其特色专业及特色课程(群)加以规划、整理和总结，更新教学内容、改革课程体系，建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上，经教育部相关教学指导委员会专家的指导和建议，清华大学出版社在多个领域精选各高校的特色课程，分别规划出版系列教材，以配合“质量工程”的实施，满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作，提高教学质量的若干意见》精神，紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”，在有关专家、教授的倡议和有关部门的大力支持下，我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”)，旨在配合教育部制定精品课程教材的出版规划，讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师，其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求，“编委会”一致认为，精品课程的建设工作从开始就要坚持高标准、严要求，处于一个比较高的起点上；精品课程教材应该能够反映各高校教学改革与课程建设的需要，要有特色风格、有创新性(新体系、新内容、新手段、新思路，教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展，顺应并符合 21 世纪教学发展的规律，代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐)，经“编委会”认真评审，最后由清华大学出版

社审定出版。

目前，针对计算机类和电子信息类相关专业成立了两个“编委会”，即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括：

（1）21世纪高等学校规划教材·计算机应用——高等学校各类专业，特别是非计算机专业的计算机应用类教材。

（2）21世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

（3）21世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

（4）21世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

（5）21世纪高等学校规划教材·信息管理与信息系统。

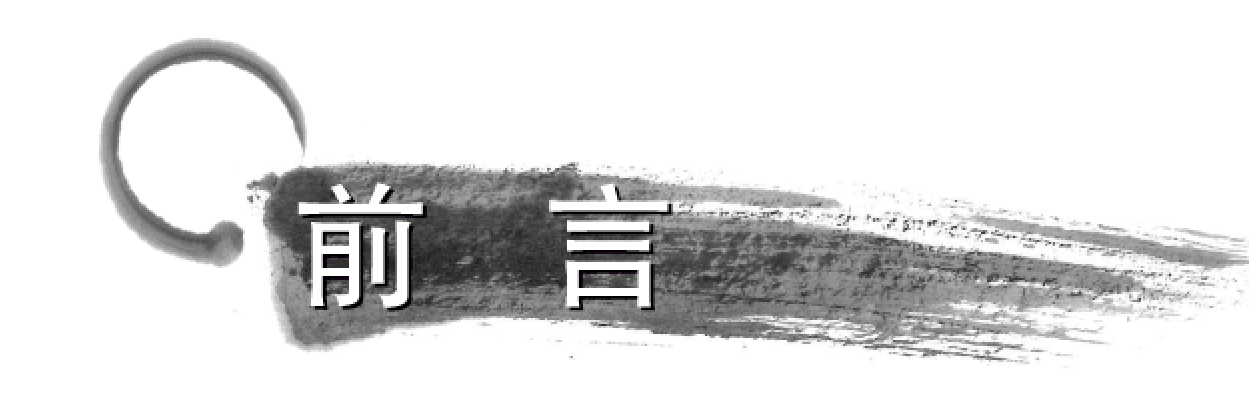
（6）21世纪高等学校规划教材·财经管理与应用。

（7）21世纪高等学校规划教材·电子商务。

（8）21世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力，在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌，为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格，这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会
联系人：魏江江
E-mail：weijj@tup.tsinghua.edu.cn



Web应用已经成为Internet上占有绝对优势的应用，从网上商城到在线游戏，从企业形象展示到社交网站，无一不体现出Web应用的强大影响力。Web应用不仅仅是在Internet上，在企业内部网络(Intranet)中，也越来越多地采用Web应用的办法完成管理与信息交流工作。因此，掌握一定的Web设计与开发的知识和技能，成为当今信息时代大专院校学生知识结构中不可或缺的一环。

作为Web设计的进阶教材，本书涉及了Web前端设计方面必备的基础知识，其中包括HTML、CSS、jQuery，图片处理软件Photoshop，动画设计工具Flash和网页设计工具Dreamweaver。

为配合本科院校的双语教学，本书特地编写了英文内容的HTML和CSS章节(第二部分Chapter2及Chapter3，对应的中文译文为第一部分的第2章和第3章)。

建议有能力的读者学习英文编写的Chapter2、Chapter3，熟练掌握Web前端设计方面的英文术语后，在资源丰富的英文社区与论坛获得帮助将不会有语言障碍。

本书强调应用。书中不仅有丰富的例子，还有实战性强的项目设计，可以使读者从学习模仿迅速过渡到实战。本书例子及综合应用实例的源代码均提供下载。

本书由吴强策划，张杰负责组稿，参与本书编著工作的作者有浙江清华长三角研究院吴强，郑州轻工业学院张杰、吉星、李建春、梁文静、张静、李勇等。其中第一部分，张静编写第1章和第7章，张杰编写第2章、第3章，吉星编写第4章，梁文静编写第5章，李建春、李勇编写第6章；第二部分由张杰编写；全书最后由吴强、李勇统一审核，河南中烟工业有限责任公司的张明明对本书的编著提了很多宝贵意见。在此，特别感谢清华大学出版社、郑州轻工业学院、浙江清华长三角研究院、河南中烟工业有限责任公司等单位对编著工作的支持。

外籍专家John Sharp审核了本书的英文内容，在此表示感谢。

由于作者水平有限，书中难免有不妥及错误之处，敬请读者批评指正。

编　者

2014年8月

于郑州



第一部分

第 二 部 分

第 一 部 分

第1章 Web基础知识

1.1 Internet 的发展

1.1.1 Internet 起源

Internet 即因特网，是全世界最大的计算机网络，其前身是美国国防部高级研究计划局（ARPA）在 20 世纪 60 年代末主持研制的 ARPANet（阿帕网）。到 20 世纪 70 年代，ARPANet 实现了在几十台计算机的网络内部通信，不同计算机网络之间仍然不能互通。为此，ARPA 又设立了新的研究项目，研究的主要内容是实现不同计算机局域网的互联互通，形成互联网（Internetwork），简称为 Internet。

1986 年，美国国家科学基金组织（NSF）建立了 NSFnet，实现了将分布在美国不同地点的 5 个为科研教育服务的超级计算机中心互联，并支持地区网络。1988 年，NSFnet 取代 ARPANet 成为 Internet 的主干网。1992 年，美国 IBM、MCI、MERIT 三家公司联合组建 ANSnet，成为 Internet 的另一个主干网，从而使 Internet 开始走向商业化。

1995 年 4 月 30 日，NSFnet 正式宣布停止运作。此时 Internet 的骨干网已经覆盖了全球 91 个国家，主机超过 400 万台。1998 年，Internet 上的用户将突破 1 亿，2000 年，全世界拥有 100 多万个网络，1 亿台主机和超过 10 亿的用户。Internet 已不再是计算机人员和军事部门进行科研的领域，而是变成了一个开发和使用信息资源的覆盖全球的信息海洋。同时，Internet 的应用也渗透到了各个领域——从学术研究到股票交易、从学校教育到娱乐游戏、从联机信息检索到在线居家购物等，Internet 已成为人类工作、生活中不能或缺的一部分。

Internet 在我国起步较晚，但发展迅速。1986 年，国内一些科研单位（如：中国科学院）开始初步接触 Internet，通过电话拨号链接到欧洲一些国家，进行国际联机数据库信息检索。1990 年实现了中国用户与 Internet 之间的电子邮件通信。目前中国和 Internet 互联的主要网络有中国教育科研网（CERNET）、中国科学技术网（CSTNET）、金桥网（GBNET），中国公用计算机互联网（Chinanet）等。前两个网络是以教育、科研服务为目的的，属于非赢利性质；后两个网络是以商业经营为目的的，又称为商业网。

1.1.2 计算机网络及其功能

现在,计算机网络的精确定义并未统一,可简单的定义为以交换共享信息为目的的多台自治计算机的互联集合。自治计算机是指网络中计算机之间不存在主从关系,各自是一个独立的工作系统;互联是指两台计算机通过通信介质连接在一起,互相交换信息。也可以将计算机网络的组成和功能定义得具体一些,即计算机网络是指将地理位置不同的多台自治计算机系统及其外部设备,通过通信介质互联,在网络操作系统、网络管理软件及网络通信协议的管理和协调下,实现资源共享和信息传递的系统。最简单的计算机网络就只有两台计算机和连接它们的一条通信线路,即两个节点和一条链路。

计算机网络的功能主要体现在以下几个方面。

(1) 信息交换

这是计算机网络最基本的功能,主要完成计算机网络中各个节点之间的系统通信,也可以称为数据通信。例如:电子邮件、发布新闻消息、电子贸易、远程电子教育等。

(2) 资源共享

资源是指构成计算机网络系统的所有要素,包括硬件资源、软件资源和数据资源,其中共享数据资源最为重要。例如计算处理能力、大容量磁盘、高速打印机、绘图仪、通信线路、数据库、文件和其他计算机上的有关信息,网络上的计算机可以共享整个网络的资源。

(3) 分布式处理

一项复杂的任务可以划分成许多部分,由网络内各计算机分别协作并执行完成有关部分,使整个系统的性能大为增强。

(4) 集中管理

计算机网络技术的发展和应用,已使得现代办公、经营管理的模式发生了很大的变化。目前,已经有了许多 MIS 系统、OA 系统等,通过这些系统可以实现日常工作的集中管理,提高工作效率,增加经济效益。

(5) 远程传输

在计算机网络中,分布在世界各地的用户可以互相传输数据信息,互相交流,协同工作。

(6) 负载均衡

负载均衡是指工作被均匀地分配给网络上的各台计算机。网络控制中心负责分配和检测,当某台计算机负载过重时,系统会自动转移部分工作到负载较轻的计算机中去处理。

1.1.3 分组交换思想

分组交换技术(包交换)是在传输线路质量不高、网络技术手段还较单一的情况下,产生的一种信息交换技术。分组交换在传输信息时,将传送的数据划分成分组,分组是一定长度二进制信息,然后通过在计算机和终端之间发送分组实现计算机与计算机之间的通信。分组交换在每个分组的前面加上一个分组头,用以指明该分组发往何地址,然后由交换机根据每个分组的地址标志,将它们转发至目的地,这一过程称为分组交换。

进行分组交换的通信网称为分组交换网。从交换技术的发展历史看,数据交换经历了电路交换、报文交换、分组交换和综合业务数字交换的发展过程。分组交换实质上是在“存

储—转发”基础上发展起来的。它兼有电路交换和报文交换的优点。分组交换在线路上采用动态复用技术传送按一定长度分割为许多小段的数据——分组。每个分组标识后，在一条物理线路上采用动态复用的技术，同时传送多个数据分组。把来自用户发端的数据暂存在交换机的存储器内，接着在网内转发。到达接收端，再去掉分组头将各数据字段按顺序重新装配成完整的报文。分组交换比电路交换的电路利用率高，比报文交换的传输时延小，交互性好。

分组交换网是继电路交换网和报文交换网之后一种新型交换网络，它主要用于数据通信。分组交换是一种存储转发的交换方式，它将用户的报文划分成一定长度的分组，以分组为存储转发，因此，它比电路交换的利用率高，比报文交换的时延要小，而具有实时通信的能力。分组交换利用统计时分复用原理，将一条数据链路复用成多个逻辑信道，最终构成一条主叫、被叫用户之间的信息传送通路，称之为虚电路（VC），实现数据的分组传送。

分组交换网具有如下特点：

① 分组交换具有多逻辑信道的能力，故中继线的电路利用率高；

② 可实现分组交换网上的不同码型、速率和规程之间的终端互通；

③ 由于分组交换具有差错检测和纠正的能力，故电路传送的误码率极小；

④ 分组交换的网络管理功能强。

分组交换的基本业务有交换虚电路（SVC）和永久虚电路（PVC）两种。交换虚电路如同电话电路一样，即两个数据终端要通信时先用呼叫程序建立电路（即虚电路），然后发送数据，通信结束后用拆线程序拆除虚电路。永久虚电路如同专线一样，在分组网内两个终端之间在申请合同期间提供永久逻辑连接，无须呼叫建立与拆线程序，在数据传输阶段，与交换虚电路相同。

分组交换数据网是由分组交换机、网路管理中心、远程集中器、分组装拆设备以及传输设备等组成。

1.1.4　ARPANet

ARPANet 是美国国防部高级研究计划局 DARPA（Defense Advanced Research Projects Agency）建立的，于 1969 年投入使用。建立 ARPANet 主导思想是：网络必须能够经受住故障的考验而维持正常工作。一旦发生战争，当网络的某一部分因遭受攻击而失去工作能力时，网络的其他部分应当能够维持正常通信。

1972 年，ARPANet 在首届计算机后台通信国际会议上与公众见面，并验证了分组交换技术的可行性，由此，ARPANet 成为现代计算机网络诞生的标志。1983 年，ARPANet 分裂为两部分：ARPANet 和纯军事用的 MILNET。之后，以 ARPANet 为主干网组成了网际互联网 Internet。

1.1.5　TCP/IP 结构模型

TCP/IP 参考模型如图 1.1 所示，由网络接口层、互联网层、传输层、应用层 4 层组成，每层功能如下。

网络接口层：TCP/IP 模型的最低层，负责接收从 IP 层交来的 IP 数据报并将 IP 数据

报通过低层物理网络发送出去，或者从低层物理网络上接收物理帧，抽出 IP 数据报，交给 IP 层。

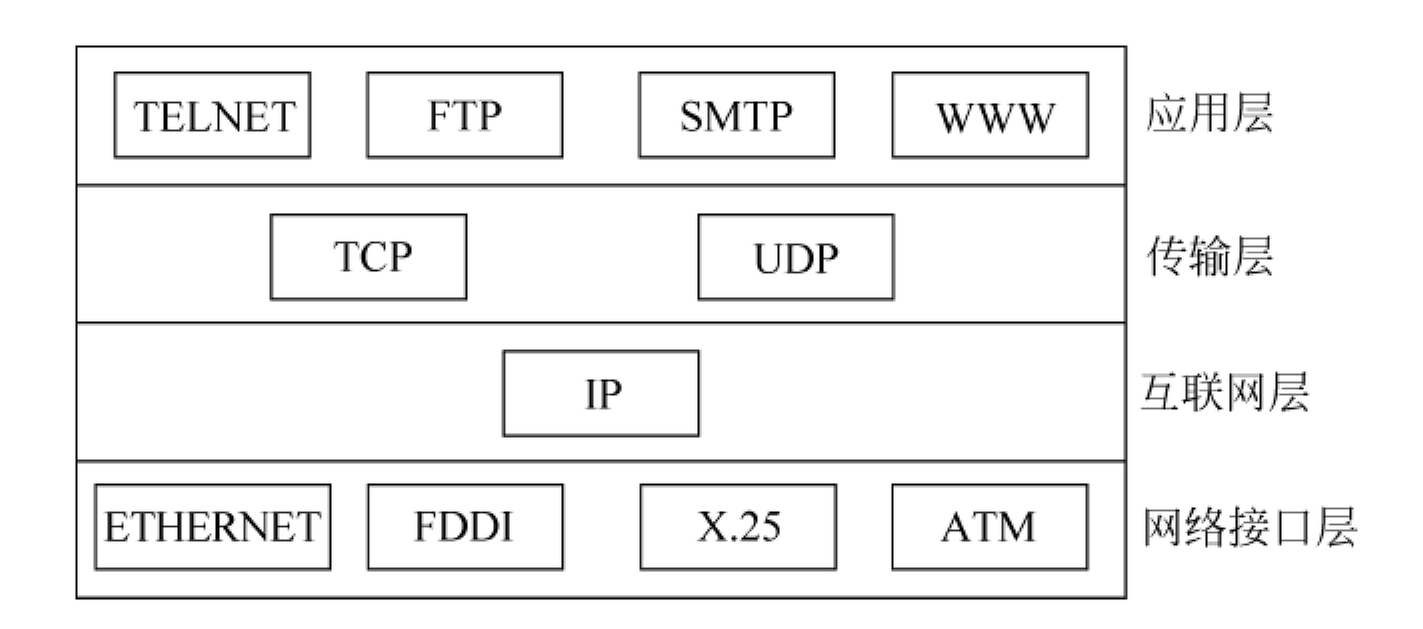


图 1.1 TCP/IP 模型

互联网层：负责相邻节点之间的数据传送。它的主要功能包括三个方面。第一，处理来自传输层的分组发送请求：将分组装入 IP 数据报，填充报头，选择去往目的节点的路径，然后将数据报发往适当的网络接口；第二，处理输入数据报：首先检查数据报的合法性，然后进行路由选择，假如该数据报已到达目的节点（本机），则去掉报头，将 IP 报文的数据部分交给相应的传输层协议；假如该数据报尚未到达目的节点，则转发该数据报；第三，处理 ICMP 报文：即处理网络的路由选择、流量控制和拥塞控制等问题。TCP/IP 网络模型的互联网层在功能上非常类似于 OSI 参考模型中的网络层。

传输层：在源节点和目的节点的两个进程实体之间提供可靠的端到端的数据传输。为保证数据传输的可靠性，传输层协议规定接收端必须发回确认，并且假定分组丢失，必须重新发送。TCP/IP 模型提供了两个传输层协议：传输控制协议 TCP 和用户数据报协议 UDP。

应用层：TCP/IP 模型没有会话层和表示层。应用层包含所有的高层协议，如虚拟终端协议（TELNET）、文件传输协议（FTP）和电子邮件协议（SMTP）等。

TCP/IP 将不同的底层物理网络、拓扑结构隐藏起来，向用户和应用程序提供通用的、统一的网络服务。这样，从用户的角度看，整个 TCP/IP 互联网就是一个统一的整体，它独立于各种具体的物理网络技术，能够向用户提供一个通用的网络服务，如图 1.2 所示。在某种意义上，可以把这个单一的网络看作一个虚拟网：在逻辑上它是独立的、统一的，在物理上它则是由不同的网络互联而成。将 TCP/IP 互联网看作单一网络的观点，极大地简化了细节，使用户极容易建立起 TCP/IP 互联网的概念。

TCP/IP 互联网还有一个基本思想：即任何一个能传输数据分组的通信系统，均可被看作是一个独立的物理网络，这些通信系统均受到互联网协议的平等对待。大到广域网、小到 LAN，甚至两台机器之间的点对点专线以及拨号电话线路都被当作网络，这就是互联网的网络对等性。网络对等性为协议设计者提供了极大的方便，大大简化了对异构网的处理。

可见，TCP/IP 网络完全撇开了底层物理网络的特性，是一个高度抽象的概念，正是这一抽象的概念，为 TCP/IP 网络赋予了巨大的灵活性和通用性。

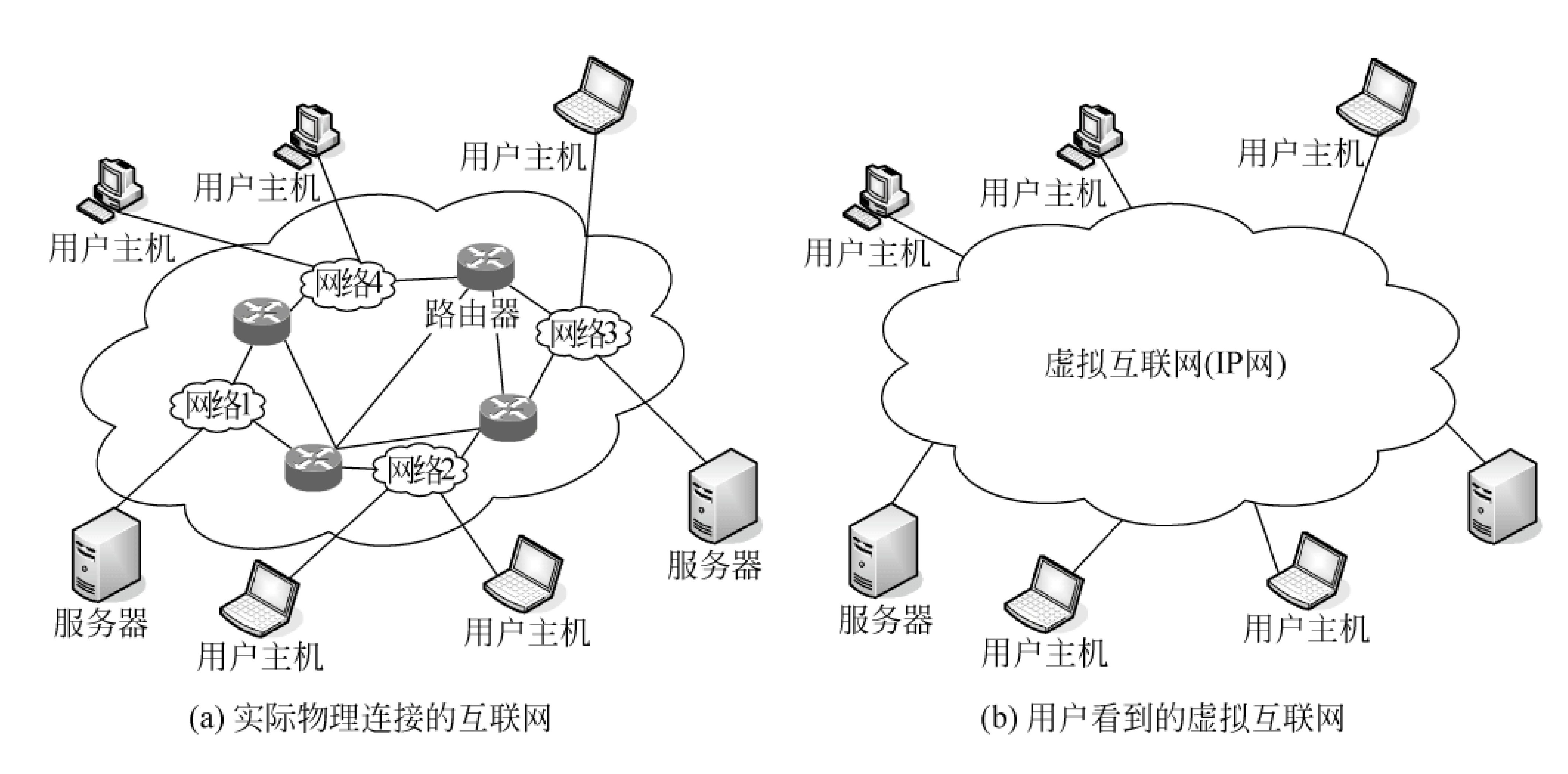


图 1.2 TCP/IP 互联网用户视图和内部结果

1.1.6 IP 地址

在 TCP/IP 体系中,IP 地址是一个最基本的概念。有关 IP 最重要的文档就是[RFC791],它很早就成为了因特网的正式标准。

1. IP 地址及其表示方法

IP 地址就是给每个连接在因特网上的主机(或路由器)分配一个在全世界范围是唯一的 32b 的标识符。IP 地址现在由因特网名字与号码指派公司 ICANN(Internet Corporation for Assigned Names and Numbers)进行分配,IP 地址的编码方法共经过了三个历史阶段,这三个阶段是:

(1) 分类 IP 地址。这是最基本的编址方法,在 1981 年就通过了相应的标准协议。

(2) 子网的划分。这是对最基本的编址方法的改进,其标准(RFC950)在 1985 年通过。

(3) 构成超网。这是比较新的无分类编址方法。1993 年提出后很快就得到推广应用。

所谓"分类的 IP 地址"就是将 IP 地址划分为若干个固定类,每一类地址都由两个固定长度的字段组成,其中一个字段是网络号 net-id,标志主机(或路由器)所连接到的网络,而另一个字段则是主机号 host-id,标志该主机(或路由器)。这种两级的 IP 地址可以记为:

```
IP 地址 = {<网络号>,<主机号>}
```

图 1.3 给出了各种 IP 地址的网络号字段和主机号字段,这里 A 类、B 类和 C 类地址是最常用的。

从图 1.3 可以看出:

(1) A 类、B 类和 C 类地址网络号字段 net-id(在网络中这个字段是灰色的)长度分别是 1、2 和 3 字节长,而在网络号字段的最前面有 1～3b 的类别比特,其数值分别规定为 0,10 和 110。

(2) A 类、B 类和 C 类地址的主机号字段长度分别为 3B、2B 和 1B。

路由器为转发分组而查找转发表时,很重要的一点就是查找路由表花费的时间要尽量

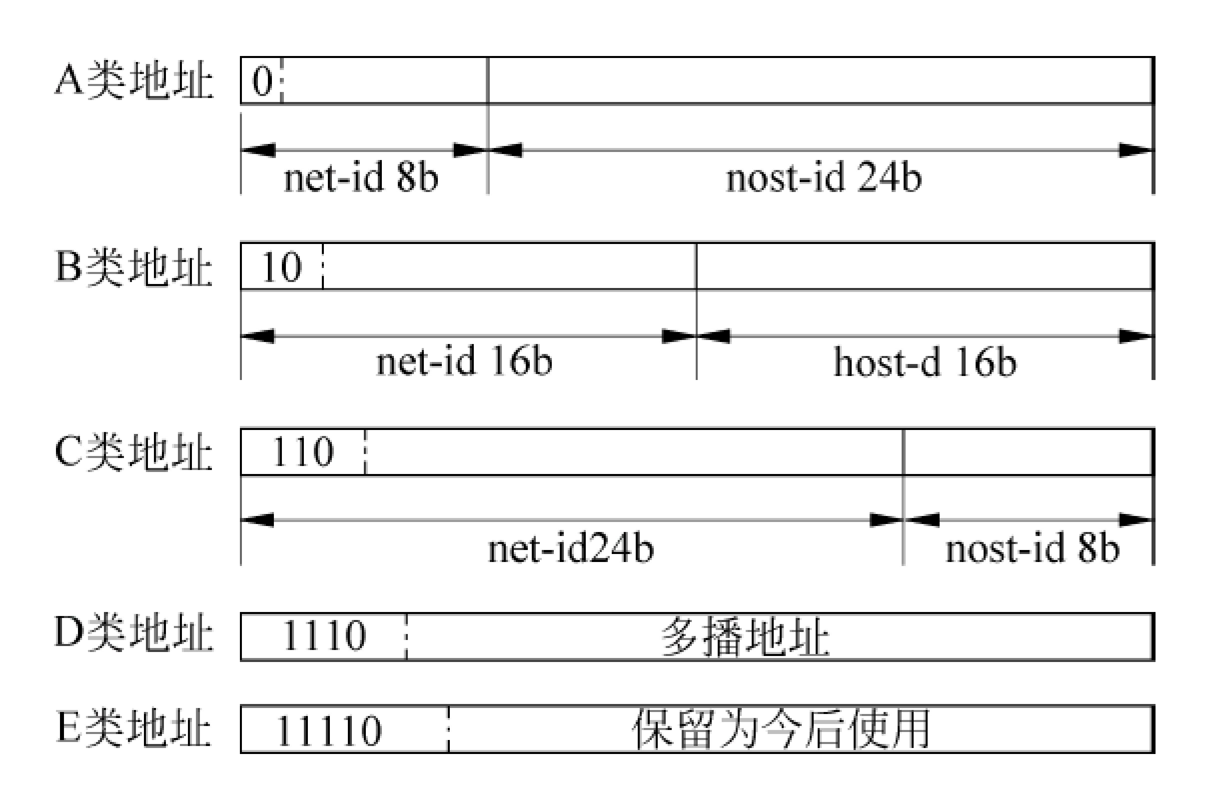


图 1.3 IP 地址中的网络号字段和主机号字段

短。如果把所有目的主机的路由全都写入到路由表中，就会有很多路由是重复的，因此没有必要这样做。我们应将重复的路由合并。路由表只使用 IP 地址中的网络号 net-id 来查找路由。只要 IP 数据报能够正确到达目的网络，就可在这个网络上直接交付目的主机而不再需要经过其他路由器进行转发了。因此，路由器转发分组的步骤如下：

(1) 先按所要找的 IP 地址中的网络号 net-id 把目的网络找到。虽然 IP 地址的网络字段有三种不同的长度，但根据 IP 地址中最前面的类别比特，就可以很容易地确定网络字段的准确字节数。

(2) 当分组到达目的网络后，利用主机号 host-id 将数据报直接交付给目的主机。

按照整数字节划分 net-id 字段，可以使路由器在收到一个分组时能够更快地将地址中的网络号提取出来。

将 IP 地址划分为三个类别，当初是这样考虑的。各种网络的差异很大，有的网络拥有很多主机，而有的网络上的主机则很少，将 IP 地址划分为 A 类、B 类和 C 类可以更好地满足不同用户的要求。A 类 IP 地址的网络号数不多，现在能够申请到的 IP 地址只有 B 类和 C 类两种。

除上述的三类 IP 地址外，还有两类使用较少的地址，即 D 类和 E 类地址。D 类地址是多播地址，E 类地址保留。

在主机或路由器中存放的 IP 地址都是 32b 的二进制代码。为了提高可读性，在写出给人看的 IP 地址时，往往每隔 8b 插入一个空格。但这样还是不方便，于是常常将 32b 的 IP 地址中的每 8b 用等效的十进制数字表示，并且在这些数字之间加上一个点，这就叫做点分十进制记法(Dotted Decimal Notation)。图 1.4 表示了这种方法，这是一个 B 类 IP 地址。显然，128.11.3.31 比 10000000 00001011 00011111 读起来要方便得多。

网络类别	最大网络数	第一个可用的网络号	最后一个可用的网络号	每个网络中最大的主机数
A	126(2^7-2)	1	126	16 777 214
B	16 384(2^{14})	128.0	191.255	65 534
C	2 097 152(2^{21})	192.0.0	223.225.255	254

图 1.4 采用点分十进制记法能够提高可读性

2. 常用的三种类别的 IP 地址

下面我们再对 A 类、B 类和 C 类地址进行较深入的讨论。

A 类地址的 net-id 字段占一个字节，只有 7 个比特可供使用(该字段的第一个比特已固定为 0)，但可提供使用的网络号是 126 个(即 2^7-2)。减 2 的原因是：第一，IP 地址中的全 0 表示“这个”。Net-id 字段为全 0 的 IP 地址是个保留地址，意思是“本网络”。第二，net-id 字段为 127(即 01111111)保留作为本地软件环回测试(Loopback Tesk)本主机之用(后面三个字节的二进制数字可任意填入，但不能都是 0 或都是 1，即除了 127.0.0.0 和 127.255.255.255 以外都可以用)。A 类地址的 host-id 字段为 3 个字节，因此每一个 A 类网络中的最大主机数是 16 777 214(即 $2^{24}-2$)。这里减 2 的原因是：全 0 的 host-id 字段表示该 IP 地址是“本主机”所连接到的单个网络地址(例如，一个主机的 IP 地址为 5.6.7.8，则该主机所在的网络地址就是 5.0.0.0)，而全 1 表示“所有的”。因此全 1 的 host-id 字段表示网络上的所有主机。

整个 A 类地址空间共有 2^{31}(即 2 147 483 648)个地址，而 IP 地址全部的地址空间共有 2^{32}(即 4 294 967 296)个地址。可见 A 类地址占有整个 IP 地址空间的 50%。

B 类地址的 net-id 字段有 2 字节，但前面两个比特(1 0)已经固定了，只剩下 14 个比特可以变化，因此 B 类地址的网络数为 16 384(即 214)。请注意，这里不存在减 2 的问题，因为 net-id 字段最前面的两个比特(1 0)使得后面的 14 个比特无论怎样排列也不可能出现使整个 2 字节 net-id 字段成为全 0 或全 1。B 类地址的每一个网络上的最大主机数是 65 534(即 $2^{16}-2$)。这里需要减 2 是因为要扣除全 0 和全 1 的主机号。整个 B 类地址空间共有 1 073 741 824(即 2^{30})个地址，占整个 IP 地址空间的 25%。

C 类地址有 3 个字节的 net-id 字段，最前面的 3 个比特是(1 1 0)，还有 21 个比特可以变化，因此 C 类地址的网络总数是 2 097 152(即 2^{21}，这里不需要减 2)。每一个 C 类地址的最大主机数是 254(即 2^8-2)。整个 C 类地址空间共有 536 870 912(即 2^{29})个地址，占整个 IP 地址的 12.5%，这样就可得出表 1.1 所示的 IP 地址的使用范围。

表 1.1　IP 地址的使用范围

网络类别	最大网络数	第一个可用的网络号	最后一个可用的网络号	每个网络中最大的主机数
A	126(2^7-2)	1	126	16 777 214
B	16 384(2^{14})	128.0	191.255	65 534
C	2 097 152(2^{21})	192.0.0	223.225.255	254

IP 地址具有以下一些重要的特点：

(1) 每一个 IP 地址都由网络号和主机号两部分组成。从这个意义上说，IP 地址是一种分层的地址结构。分两个层次的好处是：第一，IP 地址管理机构在分配 IP 地址时只分配网络号(第一级)，而剩下的主机号(第二级)则由得到该网络号的单位自行分配。这样就方便了 IP 地址的管理。第二，路由器仅根据目的主机所连接的网络号来转发分组(而不考虑目的主机号)，这样就可以使路由表中的项目数大幅度减少，从而减少了路由表所占的存储空间。

(2) 实际上 IP 地址是标志一个主机(或路由器)和一条链路的接口。当一个主机同时连接到两个网络上时,该主机就必须同时具有两个相应的 IP 地址,其网络号 net-id 必须是不同的。这种主机称为多接口主机(Multihomed Host)。由于一个路由器至少应当连接到两个网络,因此一个路由器至少应当有两个不同的 IP 地址。

(3) 按照因特网的观点,用转发器或网桥连接起来的若干个局域网仍为一个网络,因为这些局域网都具有同样的网络号 net-id。

(4) 在 IP 地址中,所有分配到网络号 net-id 的网络都是平等的。

图 1.5 画出了 3 个局域网(LAN1、LAN2 和 LAN3)通过 3 个路由器(R1、R2、和 R3)互连起来所构成的一个互联网(此互联网用虚线圆角方框表示)。其中局域网 LAN2 是两个网段通过网桥 B 互连的。图中的小圆圈表示需要有一个 IP 地址。

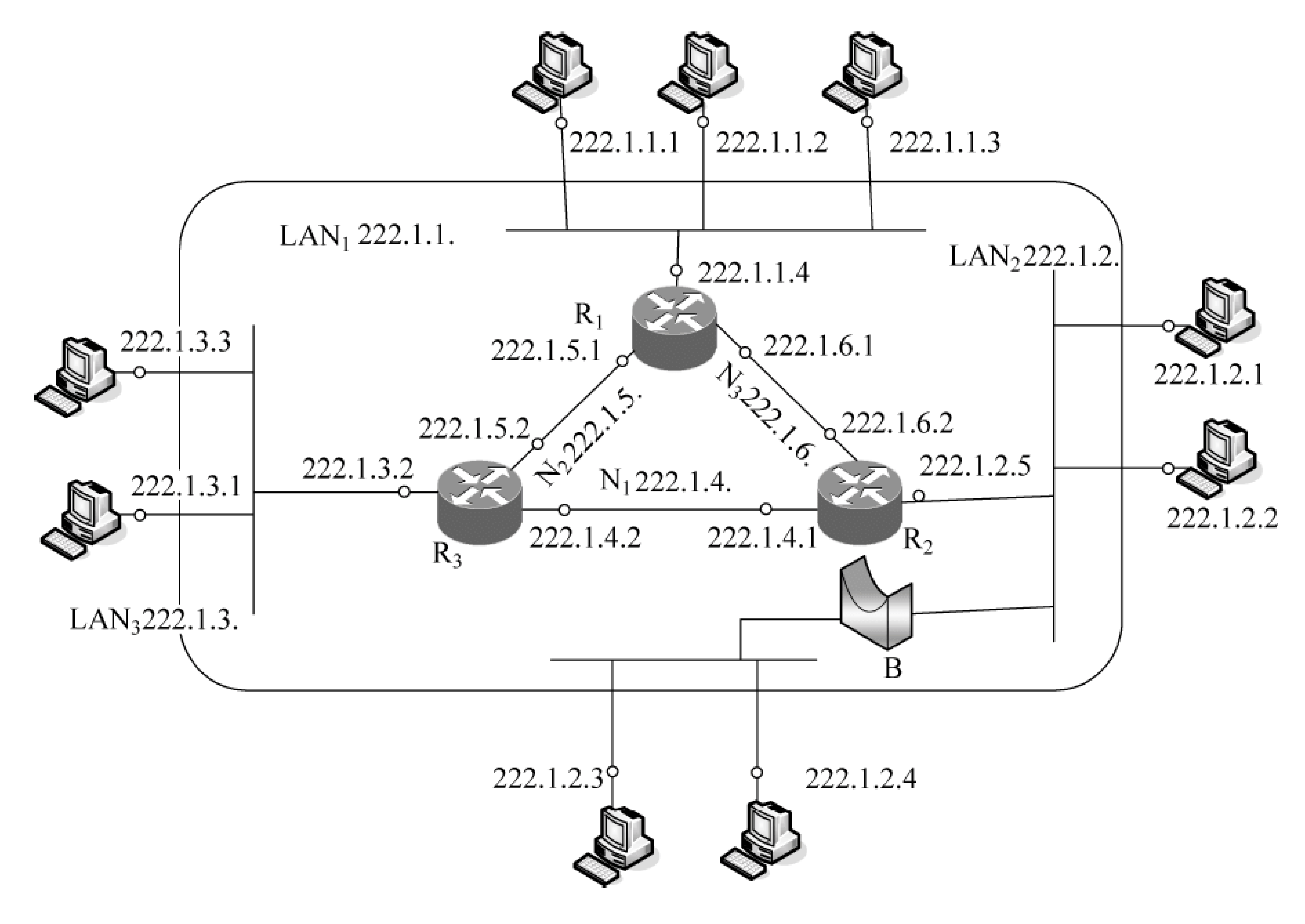


图 1.5 互联网中的 IP 地址

注意:

(1) 在同一个局域网上的主机或路由器的 IP 地址中的网络号必须是一样的。图 1.5 中所示的网络号就是 IP 地址中的 net-id,这也是文献中常见的一种表示方法。另一种表示方法是用主机号 host-id 为全 0 的网络 IP 地址。

(2) 用网桥(它只在链路层工作)互连的网段仍然是一个局域网,只能有一个网络号。路由器总是具有两个或两个以上的 IP 地址。即路由器的每一个接口都有一个不同网络号的 IP 地址。

(3) 当两个路由器直接相连时,在连线两端的接口处,可以指明也可以不指明 IP 地址。如指明了 IP 地址,则这一段连线就构成了一种只包含一段线路的特殊"网络"(如图中的 N1,N2 和 N3),之所以叫做"网络"是因为它有 IP 地址,但为了节省 IP 地址资源,对于这种由一段连线构成的特殊"网络"现在也常常不指明 IP 地址。

1.1.7 DNS 域名系统

早期的因特网使用平坦(Flat)名字空间,即主机名(Host Name)没有层次关系,其优点是名字简短。在 ARPANET 时代,整个网络上只有数百台主机,主机中都保存一个 hosts 的文件,其中列出了所有主机名字和相应的 IP 地址,至今在一些网络操作系统中都还保存了这种机制。随着因特网上的用户数急剧增加,这种平坦(Flat)的名字空间已经不能满足需要,因此 DNS 域名采用层次化的名字空间。采用这种命名方法,任何一台连接在因特网上的主机或路由器,都有一个唯一的层次结构的名字,即域名(Domain Name)。

域名的结构由若干个分量组成,各分量之间用点隔开:

…….三级域名.二级域名.顶级域名

各分量代表不同级别的域名。每一级的域名都由英文字母和数字组成(不超过 63 个字符,并忽略大小写),级别最低的域名写在最左边,而级别最高的域名称为"顶级域名",写在最右边。完整的域名不超过 255 个字符。域名只是一个逻辑概念,并不代表计算机的物理地点。域名中的点与 IP 地址中的点并无对应关系。

假设郑州轻工业学院的一台服务器 www.zzuli.edu.cn,其中 cn 是顶级域名;edu 是二级域名;zzuli 是三级域名。该学院拥有 zzuli.edu.cn 域名及其以下的所有名字空间,其中 www 是该学院是一台服务器的主机名。

域名有了层次结构,就可以采用层次结构管理的方法对 Internet 上的域名进行管理,不需要每个 DNS 服务器都知道 Internet 上所有的域名,一个名称服务器可以把它的一部分名称服务"委托"给子服务器,从而实现名字空间的层次结构。

因特网全球域名服务系统(InterDNS)由 InterNIC 管理,InterDNS 名字空间是一种树状结构,它指定了一个用于组织名称的结构化的层次式域名空间。树状结构的国际互联网的域名体系如图 1.6 所示。

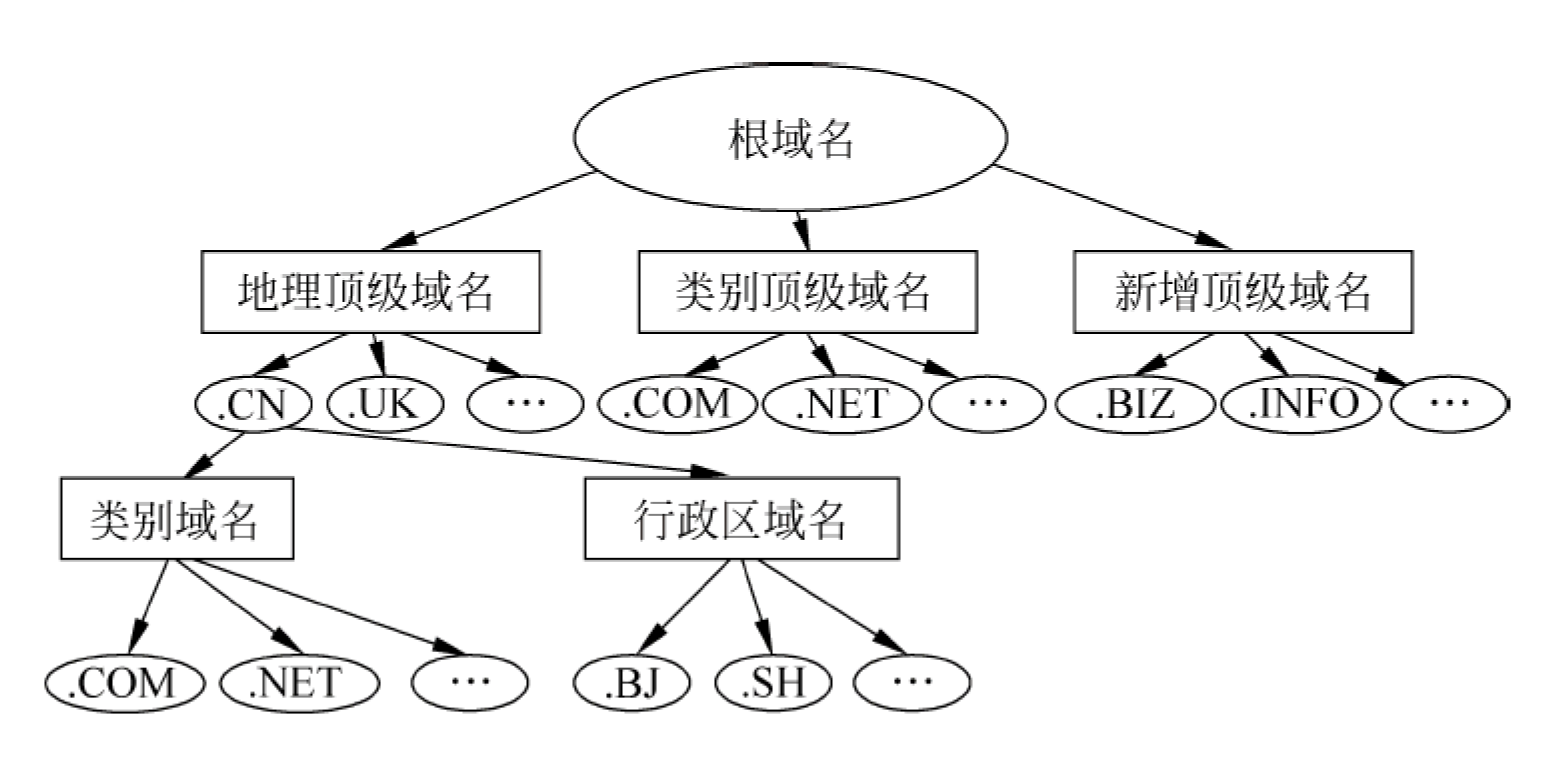


图 1.6　国际互联网域名体系

顶级域名由 InterNIC 或相关的管理机构进行管理,目前顶级域名共分为三类:

第一类是类别顶级域名,共有 7 个,也就是现在通常说的国际域名。由于 Internet 最初是发源于美国,因此最早的域名并无国家标识,人们按用途把它们分为几个大类,它们分别以不同的后缀结尾:.com(用于商业公司);.net(用于网络服务);.org(用于组织协会等);.gov(用于政府部门);.edu(用于教育机构);.mil(用于军事领域);.int(用于国际组织)。最初的域名体系也主要供美国使用,因此美国的企业、机构、政府部门等所用的都是"国际域

名”,随着 Internet 向全世界的发展,.edu、.gov、.mil 一般只被美国专用外,另外三类常用的域名.com、.org、.net 则成为全世界通用,因此这类域名通常称为“国际域名”,直到现在仍然为世界各国所使用。上述 7 个顶级域名由 InterNIC 负责管理。

第二类是地理顶级域名,共有 243 个国家和地区的代码,例如.cn 代表中国,.uk 代表英国。这样以.cn 为后缀的域名就相应地叫做“国内域名”。由各国的互联网管理机构自行管理。

与国际域名的后缀命名类似,在.cn 顶级域名下也分设了不同意义的二级域,主要包括类别域和行政区域,这样就是通常说的二级域名:

类别域名是依照申请机构的性质划分出来的域名,具体包括:

AC 科研机构;

COM 工、商、金融等企业;

EDU 教育机构;

GOV 政府部门;

NET 互联网络、接入网络的信息中心(NIC)和运行中心(NOC);

ORG 各种非盈利性的组织。

行政区划域名是按照中国的各个行政区划划分而成的,其划分标准依照原国家技术监督局发布的国家标准而定,包括“行政区域名”34 个,适用于我国的各省、自治区、直辖市。例如北京的机构可以选择如 cnnic.bj.cn 的域名。

中国行政区域名举例:

BJ-北京市;	SH-上海市;	TJ-天津市;
CQ-重庆市;	HE-河北省;	SX-山西省;
NM-内蒙古自治区;	LN-辽宁省;	JL-吉林省;
HL-黑龙江省;	JS-江苏省;	ZJ-浙江省;
AH-安徽省;	FJ-福建省;	JX-江西省;
SD-山东省;	HA-河南省;	TW-台湾省;
HK-香港;	MO-澳门。	……

第三类顶级域名,也就是所谓的“新顶级域名”,是 ICANN 根据互联网发展需要,在 2000 年 11 月做出决议,从 2001 年开始使用的国际顶级域名,也包含 7 类:biz,info,name,pro,aero,coop,museum。其中前 4 个是非限制性域,后 3 个是限制性域,如 aero 需是航空业公司注册,museum 需是博物馆注册,coop 需是集体企业(非投资人控制,无须利润最大化)注册。这 7 个顶级域名的含义和注册管理机构如下:

.aero,航空运输业专用,由比利时国际航空通信技术协会(SITA)负责;

.biz,可以替代.com 的通用域名,监督机构是 JVTeam;

.coop,商业合作社专用,由位于华盛顿的美国全国合作商业协会(NCBA)负责管理;

.info,可以替代.net 的通用域名,由 19 个因特网域名注册公司联合成立的 Afilias 负责;

.museum,博物馆专用,由博物馆域名管理协会(MDMA)监督;

.name,个人网站的专用域名,由英国的“环球姓名注册”(Globe Name Registry)负责;

.pro,医生和律师等职业专用,监督机构是爱尔兰都柏林的一家网络域名公司“职业注册”(Registry Pro)。

1.2　万维网与浏览器

1.2.1　万维网

万维网(World Wide Web)是环球信息网,缩写为 WWW,也可以简称为 Web。万维网是一个资料空间。在这个空间中;一样有用的事物,称为一样“资源”;并且由一个全域“统一资源标识符”(URL)标识。这些资源通过超文本传输协议(Hypertext Transfer Protocol)传送给使用者,而后者通过单击链接来获得资源。从另一个观点来看,万维网是一个透过网络存取的互连超文件(Interlinked Hypertext Document)系统。

万维网由欧洲原子核研究委员会 CERN(the European Laboratory for Particle Physics,CERN 是法文缩写)委员蒂姆·伯纳斯(Tim Berners-Lee)于 1989 年 3 月提出的一种基于 Internet 的信息服务系统,目的是使分散在欧洲各国的物理学家能够通过计算机网络合作进行科学研究,使分散在各地物理实验室的信息能被共享。1991 年,CERN 向世界公布了 WWW 技术,WWW 的出现立即在世界上引起轰动。万维网联盟(World Wide Web Consortium,W3C),又称 W3C 理事会。1994 年 10 月在拥有“世界理工大学之最”称号的麻省理工学院(MIT)计算机科学实验室成立,建立者是万维网的发明者蒂姆·伯纳斯·李。

查看万维网必须使用相应的软件——浏览器(Browser)。第一个图形化 Web 浏览器是美国国家超级计算机应用中心(NCSA)的马克·安德森(Mark Anderson)于 1993 年 2 月在 Windows 环境下开发出的 MOSAIC 浏览器。MOSAIC 基本上类似于现在的浏览器软件,它解决了远程信息服务中的文字显示、数据连接以及图像传递等问题,这比起最初的基于字符界面的浏览器来说是一个巨大的进步。MOSAIC 的出现和广泛使用大大推动了万维网的发展,使万维网迅速风靡全世界。1995 年著名的 Netscape Navigator 浏览器上市。此后微软公司推出了 Internet Explorer(“探路者”,通常简称 IE),并在它的 Windows 操作系统中捆绑发布,成为最主要的 Web 浏览器。

万维网的出现,使因特网成为普通人也能利用的信息资源库,因特网上的 Web 网站点数呈指数规律增长。据统计,在 1998 年,万维网的通信量已超过整个因特网上通信量的 75%今天,Web 已经是 Internet 中被使用最多的部分,以至于给人的感觉是 Web 就是 Internet。因此,万维网的出现是因特网发展中一个非常重要的里程碑。

1.2.2　万维网分布式服务特点

万维网用链接的方法将不同的信息资源连接到一起,使用户能非常方便地从因特网上的一个站点访问另一个站点(也就是所谓的“链接到另一个站点”),从而主动地按需获取丰富的信息。图 1.7 展示了万维网站点之间的典型链接方式。

图中的万维网站点可以相隔数千公里,每一个万维网站点都存放着许多文档,在这些文档中有一些地方的文字是用区别于其他文字的方式显示的(当我们将鼠标移动到这些地方时,鼠标的箭头就变成了一只手的形状),表明这些地方有一个链接,这些文档相互链接,成为一个“网”状结构。用户只要单击鼠标,就可以通过一个文档链接到可能相隔很远的另一

个文档,浏览到世界任何地方的万维网资源。

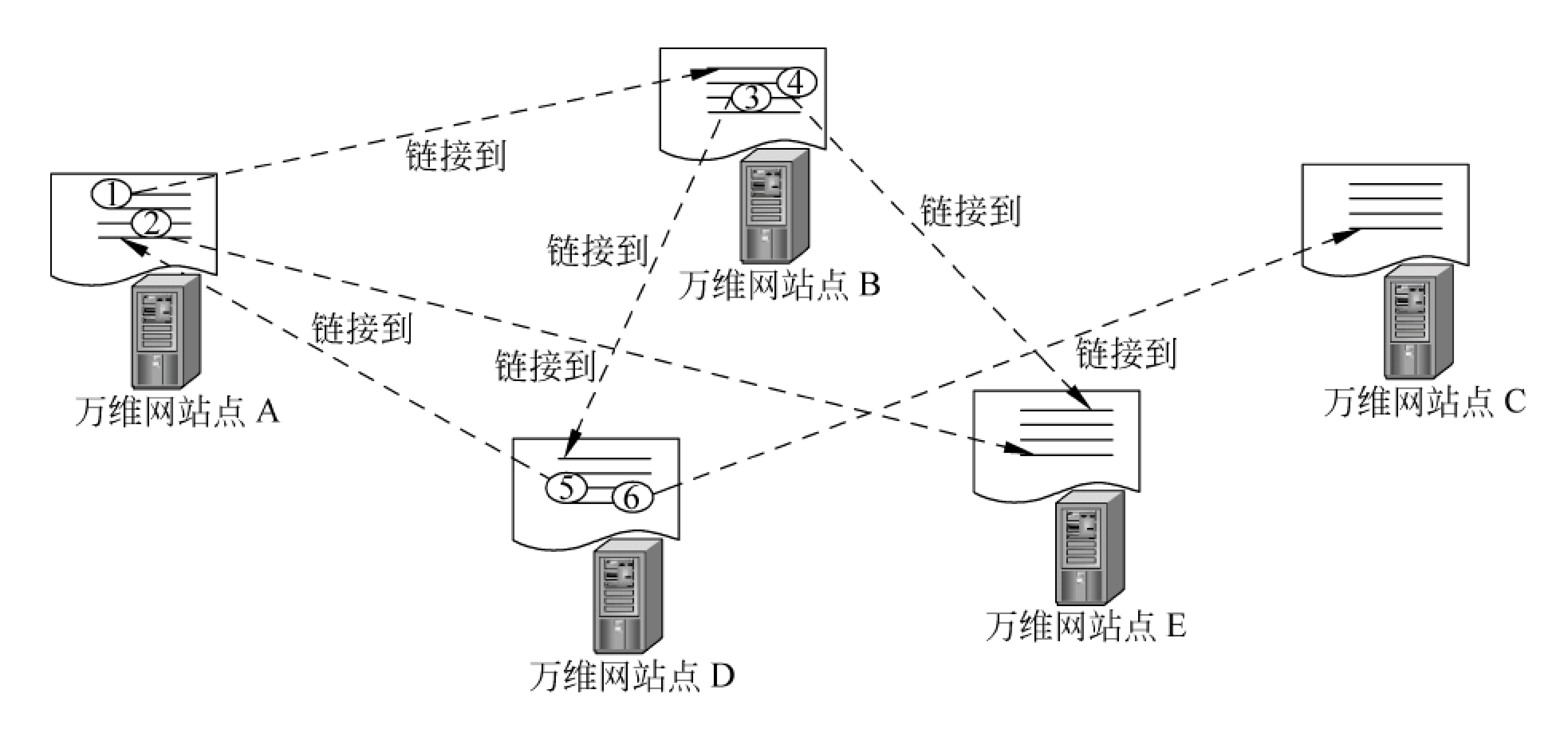


图 1.7　万维网站点之间的链接

1.2.3　万维网的工作方式

WWW 系统的基本组成包括服务器、浏览器、网页和传输协议。

万维网文档所驻留的计算机运行着万维网服务程序,因此这个计算机被称为万维网服务器或 Web 服务器,是网站的主要组成部分。

浏览器是在用户计算机上的万维网客户程序,它的作用是为用户提供一个人机接口,即操作界面,用户通过浏览器输入需要访问网页的网址,浏览器则将用户输入的网址提交给网址中指定的网站。浏览器收到网站传送过来的网页后,加以解释、执行,并将结果在屏幕上显示出来,成为用户平常所见的漂亮网页。

网页由 HTML(Hyper Text Markup Language)语言编写而成,也可称为 HTML 文档,网页中可以嵌入文字、图片、声音、视频等丰富的多媒体信息。网页之间通过超链接链接到一块,超链接就是网页的地址,因此网页可以理解为超文本系统中的一个节点。网页存放在网站服务器上,其扩展名一般是:HTM、HTML、ASP、PHP 等。

1.3　C/S 与 B/S 结构

1.3.1　C/S 结构

C/S(Client/Server)结构,即大家熟知的客户机和服务器结构。它是软件系统体系结构,通过它可以充分利用两端硬件环境的优势,将任务合理分配到 Client 端和 Server 端来实现,降低了系统的通信开销。目前大多数应用软件系统都是 Client/Server 形式的两层结构,由于现在的软件应用系统正在向分布式的 Web 应用发展,Web 和 Client/Server 应用都可以进行同样的业务处理,应用不同的模块共享逻辑组件。因此,内部的和外部的用户都可以访问新的和现有的应用系统,通过现有应用系统中的逻辑可以扩展出新的应用系统,这也是目前应用系统的发展方向。

C/S 结构的基本原则是将计算机应用任务分解成多个子任务,由多台计算机分工完成,

即采用“功能分布”原则。客户端完成数据处理，数据表示以及用户接口功能；服务器端完成 DBMS 的核心功能。C/S 结构的优点是能充分发挥客户端 PC 的处理能力，很多工作可以在客户端处理后再提交给服务器。对应的优点就是客户端响应速度快。缺点主要有：①只适用于局域网。而随着互联网的飞速发展，移动办公和分布式办公越来越普及，这需要系统具有扩展性。这种方式远程访问需要专门的技术，同时要对系统进行专门的设计来处理分布式的数据。②客户端需要安装专用的客户端软件。首先涉及到安装的工作量，其次任何一台电脑出问题，如病毒、硬件损坏，都需要进行安装或维护。特别是有很多分部或专卖店的情况，不是工作量的问题，而是路程的问题。还有，系统软件升级时，每一台客户机需要重新安装，其维护和升级成本非常高。③对客户端的操作系统一般会有限制。可能适应于 Windows 7、Windows 8 或 Windows XP，却不适用于微软新的操作系统，更不用说 Linux、UNIX 等。

C/S 架构软件的优势是应用服务器运行数据负荷较轻，数据的储存管理功能较为透明。劣势是高昂的维护成本、大投资，以及需要针对不同的操作系统开发不同版本的软件，由于产品的更新换代十分快，高代价和低效率已经不适应工作需要。

1.3.2　B/S 结构

B/S(Browser/Server，浏览器/服务器模式)结构是 Web 兴起后的一种网络结构模式，Web 浏览器是客户端最主要的应用软件。这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。客户机上只要安装一个浏览器(Browser)，如 Netscape Navigator 或 Internet Explorer，服务器安装 Oracle、Sybase、Informix 或 SQL Server 等数据库。浏览器通过 Web Server 同数据库进行数据交互。

B/S 最大的优点就是可以在任何地方进行操作而不用安装任何专门的软件，只要有一台能上网的电脑就能使用，客户端零维护，系统的扩展非常容易。B/S 结构的使用越来越多，特别是由需求推动了 AJAX 技术的发展，它的程序也能在客户端电脑上进行部分处理，从而大大地减轻了服务器的负担；并增加了交互性，能进行局部实时刷新。

B/S 架构软件的优势是维护和升级方式简单，成本降低，选择更多。劣势是应用服务器运行数据负荷较重。由于 B/S 架构管理软件只安装在服务器端(Server)上，网络管理人员只需要管理服务器就行了，用户界面主要事务逻辑在服务器(Server)端完全通过 WWW 浏览器实现，极少部分事务逻辑在前端(Browser)实现，所有的客户端只有浏览器，网络管理人员只需要做硬件维护。但是，应用服务器运行数据负荷较重，一旦发生服务器“崩溃”等问题，后果不堪设想。因此，许多单位都备有数据库存储服务器，以防万一。

C/S 结构与 B/S 的区别总结起来有以下几点。

(1) 硬件环境不同：C/S 一般建立在专用的网络上，小范围里的网络环境，局域网之间再通过专门服务器提供连接和数据交换服务。B/S 建立在广域网之上，比 C/S 适应环境的能力更强，一般只需要有操作系统和浏览器。

(2) 安全要求不同：C/S 一般面向相对固定的用户群，对信息安全的控制能力很强．一般高度机密的信息系统采用 C/S 结构适宜。B/S 建立在广域网之上，对安全的控制能力相对弱，面向不可知的用户群。

(3) 程序架构不同：C/S 程序可以更加注重流程，对权限多层次校验，对系统运行速度

考虑较少。B/S对安全以及访问速度要有多重考虑。

(4) 软件重用不同：C/S程序出于整体性考虑，构件的重用性不如在B/S环境下的构件的重用性好。B/S要求构件功能相对独立性强，能够相对较好的重用。

(5) 系统维护不同：C/S程序由于整体性要求，处理出现的问题以及系统升级都比较困难。B/S由构件组成，个别构件的更换就可以实现系统的无缝升级，系统维护开销小，而且用户从网上自己下载安装就可以实现升级。

(6) 用户接口不同：C/S多建立在Windows平台上，表现方法有限，对程序员普遍要求较高。B/S建立在浏览器上，有更加丰富和生动的表现方式与用户交流，并且使用难度减低，开发成本减少。

(7) 信息流不同：C/S程序一般是典型的中央集权的机械式处理，交互性相对较低。B/S信息流向可变化，更像交易中心。

1.4 互联网新技术及应用

1.4.1 IPv6

IPv6(Internet Protocol Version 6)是下一代互联网协议，是用来替代现行的IP(IPv4)协议的一种新的IP协议。

今天的互联网大多数应用的是IPv4协议，IPv4协议已经使用了20多年，在这20多年的应用中，IPv4获得了巨大的成功，同时随着应用范围的扩大，它也面临着越来越不容忽视的危机(例如地址匮乏)。IPv6是为了解决IPv4所存在的一些问题和不足而提出的，同时它还在许多方面提出了改进。

经过一个较长的IPv4和IPv6共存的时期，IPv6最终会完全取代IPv4在互联网上占据统治地位。IPv6所引进的主要变化如下：

(1) 更大的地址空间。IPv6将地址从IPv4的32b增大到了128b，使地址空间增大了296倍。这样大的地址空间在可预见的将来是不会用完的。

(2) 扩展的地址层次结构。IPv6由于地址空间很大，因此可以划分为更多的层次。

(3) 灵活的首部格式。IPv6数据报的首部和IPv4的并不兼容。IPv6定义了许多可选的扩展首部，不仅可提供比IPv4更多的功能，而且还可提高路由器的处理速率，这是因为路由器对扩展首部不进行处理(除逐跳扩展首部外)。

(4) 改进的选项。IPv6允许数据报包含有选项的控制信息，因而可以包含一些新的选项(IPv4所规定的选项是固定不变的)。

(5) 允许协议继续扩充。这一点很重要，因为技术总是在不断地发展(如网络硬件的更新)，而新的应用也还会出现(IPv4的功能是固定不变的)。

(6) 支持即插即用(即自动配置)。

(7) 支持资源的预分配。IPv6支持实时视像等要求保证一定的带宽和时延的应用。

在IPv6中，每个地址占128b，地址空间大于3.4×1038。如果整个地球表面(包括陆地和水面)都覆盖着计算机，那么IPv6允许每平方米拥有7×1023个IP地址。如果地址分配速率是每微秒分配100万个地址，则需要1019年的时间才能将所有可能的地址分配完毕。

可见在想象得到的将来，IPv6 的地址空间是不可能用完的。

巨大的地址范围还必须使维护互联网的人易于阅读和操纵这些地址。IPv4 所用的点分十进制记法现在也不够方便了。例如，一个用点分十进制记法的 128b 的地址：

104.230.140.100.255.255.255.255.0.0.17.128.150.10.255.255

为了使地址再稍简洁些，IPv6 使用冒号十六进制记法（colon hexadecimal notation，简写为 colon hex），它把每个 16b 的值用十六进制值表示，每个值之间用冒号分隔。例如，如果前面所给的点分十进制数记法的值改为冒号十六进制记法，就变成了：

68E6:8C64:FFFF:FFFF:0:1180:960A:FFFF

这里将 0000 中的前三个 0 省略了。例如，三个 0 后面一个 F（000F）可写为 F。冒号十六进制记法还包含两个技术使它尤其有用。首先，冒号十六进制记法可以允许零压缩（Zero Compression），即一连串连续的零可以为一对冒号所取代，例如：

FF05:0:0:0:0:0:0:B3 可以写成 FF05::B3

为了保证零压缩有一个不含混的解释，规定在任一地址中只能使用一次零压缩。该技术对已建议的分配策略特别有用，因为会有许多地址包含连续的零串。

其次，冒号十六进制记法可结合点分十进制记法的后缀。下面会看到这种结合在 IPv4 向 IPv6 的转换阶段特别有用。例如，0:0:0:0:0:0:128.10.2.1 是一个合法的冒号十六进制记法。

请注意，在这种记法中，虽然为冒号所分隔的每个值是一个 16b 的量，但每个点分十进制部分的值则指明一个字节（8b）的值。再使用零压缩即可得：::128.10.2.1。

CIDR 的斜线表示法仍然可用。例如，60b 的前缀 12AB00000000CD3（十六进制表示的 15 个字符，每个字符代表 4b）可记为：

12AB:0000:0000:CD30:0000:0000:0000:0000/60

12AB::CD30:0:0:0:0/60

12AB:0:0:CD30::/60

IPv6 不可能立刻替代 IPv4，因此在相当一段时间内 IPv4 和 IPv6 会共存在一个环境中。要提供平稳的转换过程，使得对现有的使用者影响最小，就需要有良好的转换机制。目前，IETF 推荐了双协议栈、隧道技术以及 NAT 等转换机制。

1.4.2 物联网

物联网（the Internet of Things）是新一代信息技术的重要组成部分。国际电信联盟（ITU）对物联网定义为："通过二维码识读设备、射频识别（RFID）装置、红外感应器、全球定位系统和激光扫描器等信息传感设备，按约定的协议，把任何物品与互联网相连接，进行信息交换和通信，以实现智能化识别、定位、跟踪、监控和管理的一种网络。"中国物联网校企联盟将物联网的定义为当下几乎所有技术与计算机、互联网技术的结合，实现物体与物体之间环境以及状态信息实时的共享以及智能化的收集、传递、处理、执行。广义上说，涉及信息技术的应用，都可以纳入物联网的范畴。物联网简单理解就是"物物相连的互联网"，通过智能感知、识别技术与普适计算，广泛应用于网络的融合中，是互联网的应用拓展。与其说物联网是网络，不如说物联网是业务和应用。

物联网被称为继计算机、互联网之后世界信息产业发展的第三次浪潮。2009 年 1 月 28

日，奥巴马就任美国总统后，在一次“圆桌会议”上，IBM首席执行官彭明盛首次提出“智慧地球”这一概念。当年，美国将新能源和物联网列为振兴经济的两大重点。2009年8月，温家宝“感知中国”的讲话把我国物联网领域的研究和应用开发推向了高潮，无锡市率先建立了“感知中国”研究中心，中国科学院、运营商、多所大学建立了物联网研究院，物联网被列为国家五大新兴战略性产业之一。

应用创新是物联网发展的核心，以用户体验为核心的创新2.0是物联网发展的灵魂。创新2.0即面向知识社会的下一代创新，与创新1.0相比较主要变化体现在科技创新模式的改变，即从专业科技人员实验室研发出科技创新成果后用户被动使用到技术创新成果的最终用户直接或通过共同创新平台参与技术创新成果的研发和推广应用全过程。物联网及移动泛在技术的发展，使得技术创新形态发生转变，以用户为中心、以社会实践为舞台、以人为本的创新2.0形态正在显现，例如Web 2.0、开放源代码、自由软件以及麻省理工学院提出的微观装配实验室等。实际生活场景下的用户体验是创新2.0模式的精髓。

1.4.3 移动互联

移动互联网简单理解就是将移动通信和互联网二者结合起来，成为一体。广义的移动互联网定义是指用户可以使用手机、笔记本等移动终端通过协议接入互联网；狭义的移动互联网定义则是指用户使用手机终端通过无线通信的方式访问采用WAP的网站。

移动互联网支持多种无线接入方式，根据覆盖范围的不同，可分为无线个人局域网通信技术（Wireless Personal Area Network Communication Technologies，WPAN）接入、无线局域网（Wireless Local Area Networks，WLAN）接入、无线城域网（Wireless Metropolitan Area Network，WMAN）接入、无线广域网（Wireless Wide Area Network，WWAN）接入等。

WPAN主要用于个人区域网场合（如家庭网络），被称为接入网的“附加一公里”，蓝牙（Bluetooth）是目前最流行的WPAN技术。WLAN主要用于商务休闲和企业校园等网络环境，被广泛称为Wi-Fi（无线相容性认证）网络，覆盖范围约100m，目前处于快速发展阶段，已在机场、酒店和校园等场合得到广泛应用。WMAN是一种新兴的适合于城域接入的技术，支持中速移动，视距传输可达50km。WWAN是指利用现有移动通信网络（如3G、4G）实现互联网接入，具有网络覆盖范围广、支持高速移动性、用户接入方便等优点。

3G是指将无线通信与国际互联网等多媒体通信结合的第三代移动通信系统，支持高速数据传输。3G主要特征是可提供移动宽带多媒体业务，能够同时传送声音及数据信息，下行速度峰值理论可达3.6Mb/s（一说2.8Mb/s），上行速度峰值可达384Kb/s。目前3G存在3种标准：CDMA2000（中国电信）、WCDMA（中国联通）、TD-SCDMA（中国移动）。业界将CDMA技术作为3G的主流技术。4G描述的是相对于3G的下一代通信网络，支持100Mb/s～150Mb/s的下行网络带宽，具有通信速度快、网络频谱宽、通信灵活、智能性能高、兼容性好、高质量通信等特点。

1.4.4 云计算与大数据

简单地理解，云就是计算机网络以及互联网的一种比喻。在有关计算机网络的图中经常把电信网、互联网和底层基础设施等抽象成云。云计算（Cloud Computing）是基于互联网

的相关服务的增加、使用和交付模式，通常涉及通过互联网来提供动态易扩展且经常是虚拟化的资源。云计算可以让用户通过电脑、笔记本、手机、上网本等方式接入数据中心，按需求进行运算。

有关云计算的概念有多种说法，美国国家标准与技术研究院(NIST)把云计算定义为一种按使用量付费的模式，这种模式提供可用的、便捷的、按需的网络访问，进入可配置的计算资源共享池(资源包括网络、服务器、存储、应用软件、服务)，这些资源能够被快速提供，只需投入很少的管理工作，或与服务供应商进行很少的交互。

对于"大数据"(Big Data)研究机构 Gartner 给出的定义是："大数据"是需要新处理模式才能具有更强的决策力、洞察发现力和流程优化能力的海量、高增长率和多样化的信息资产。也可以理解为"无法在一定时间内用常规软件工具对其内容进行抓取、管理和处理的数据集合"。

大数据具有四个显著的特点：

① 数据体量巨大，达到 PB 级别。

② 数据类型繁多，包含日志、视频、图片、地理位置等信息。

③ 处理速度快，快速高效地从各种类型的巨量数据中找到高价值的信息。

④ 高的价值回报，可以合理利用数据并对其进行分析，可以得到意想不到的回报。

大数据可分成大数据技术、大数据工程、大数据科学和大数据应用等领域。其中，大数据技术和大数据应用是目前研究的热点。所谓大数据技术就是指实现从各种各样类型的数据中快速获得有价值信息的技术。大数据应用对各行业都具有极大的吸引力，也是发展趋势，但如何应用也是现阶段对各行业的极大挑战。

小结

本章主要讲述了 Web 的基础知识，包括 Internet 的发展与应用、计算机网络的定义及其功能、分组交换思想、TCP/IP 的层次结构等。重点讲述了 IP 地址的作用、构成和应用。最后介绍了万维网与浏览器，以及 C/S、B/S 结构。简单介绍了当今互联网出现的一些新技术及其应用，包括物联网、云计算和大数据等。为学习后续的 Web 知识，以及进行 Web 设计、开发与应用提供了基础的概念和思路。

习题

1. 简述 Internet 的起源与发展。
2. 简述计算机网络的含义，举例说明计算机网络的主要功能。
3. 简述 TCP/IP 的层次结构，说明 IP 地址的分类，并举例说明 IP 地址的应用。
4. 简述 C/S 结构与 B/S 的区别。
5. 简述互联网在你自身学习、生活的中的应用。

第2章 HTML

2.1 认识 HTML

2.1.1 创建第一个 HTML 网页

使用 Windows 操作系统下附带的记事本来创建第一个 HTML 文件，详细步骤如下：

(1) 打开开始菜单找到记事本：选择"开始"→"所有程序"→"附件"→"记事本"。

(2) 记事本打开，此时可以输入 HTML 代码。

(3) 显示已知文件类型的扩展名：Windows 7 的资源管理器默认会隐藏已知文件类型的扩展名。例如，文件名为 hello.html 的文件在资源管理器中显示为 hello，隐藏了它的.html 扩展名。

让 Windows 7 显示这些扩展名会减少很多困惑，所以要更改文件夹选项以便可以看到文件扩展名。

首先，在任意一个资源管理器窗口中，选中"工具"菜单中的"文件夹选项"。

下一步，在"查看"选项卡中，"高级设置"之下，拖动滚动条直到看见"隐藏已知文件类型的扩展名"，取消对这个选项的选中。

单击"确定"按钮保存这个设置，今后就可以在资源管理器中看见文件的扩展名。

(4) 把下面的代码输入到记事本的空白窗口中。

例 2.1 第一个 HTML 网页。

```
<html>
<body>

  <h1>Hello, World</h1>

</body>
</html>
```

(5) 选择"文件"→"保存"菜单项来保存所做的工作，单击"保存"按钮后会看到一个"另存为"对话框。首先，单击对话框左栏的"桌面"图标，然后在"保存类型"下拉框中选择"所有文件"(否则记事本会在文件名后面添加默认的.txt 扩展名)。输入 index.html 作为文件名并单击"保存"按钮。

(6) 在浏览器中打开这个网页文件：双击刚才创建的 index. html 文件(位于桌面)，成功！浏览器显示如图 2.1 所示的网页。

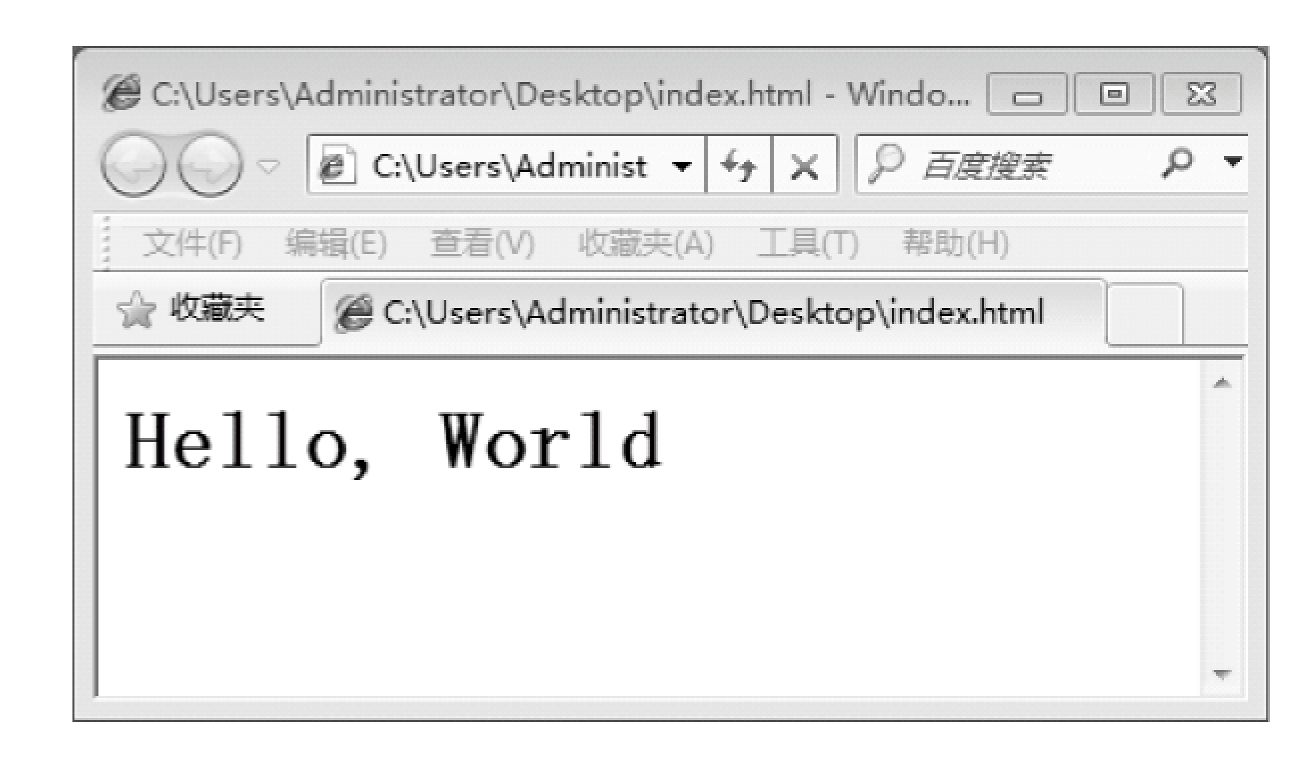


图 2.1 第一个 HTML 网页

至此，第一个 HTML 文件被创建！

提示：这是一个最简单的网页文件。它不是很完整，却最容易理解与入手。注意掌握建立网页文件与浏览网页文件的方法与步骤。网页文件的扩展名可以是 html 或者 htm，本教材一律使用. html 扩展名。

2.1.2 HTML 术语

上小节中所创建的 HTML 文件中，<html>、</html>、<h1>、<body>等是用来标记 HTML 中的文本的。现在，仔细看一看 HTML 标签在例 2.2 中是怎样工作的。

例 2.2 HTML 中的标签。

```
<h1>Hello, World</h1>
```

在例 2.2 中，<h1>和</h1>被叫做标签——<h1>是起始标签，</h1>是终止标签。通常在起始标签和终止标签中间放入一些内容。本例中，内容是“Hello，World”。标签用来告诉浏览器，它的内容“Hello，World”是一个顶级标题(也就是一级标题)。例 2.2 的全部内容叫做一个元素。在这里被称为<h1>元素。一个元素由闭合的标签和标签之间的内容组成。

用成对的标签包围内容来告诉浏览器网页的结构。标签的构成公式为：

元素＝起始标签＋内容＋终止标签

标签由标签名和包裹它的尖括号组成，也就是是<和>符号。起始标签<h1>开始一个元素，而终止标签</h1>结束一个元素。根据</h1>跟在内容后面并且它在 h1 之前有一个/判断它是一个终止标签。所有的终止标签内部都有一个/符号。

提示：本单元是第一次介绍有关 HTML 的专业术语，为方便学习与消化，本教程把专业术语按需要循序渐进地通过举例介绍给读者，后边讲到的专业术语也顺便复习了前边所讲的内容，加深印象与理解。当然，最重要的是掌握理解代码与编辑代码的能力。

2.1.3 HTML4 和 HTML5

HTML(HyperText Markup Language，超文本标记语言)是万维网的原始语言，现在提

到它通常是指它的现行版本 HTML4.01 或者简称 HTML4。HTML 最初是 SGML (Standard Generalized Markup Language,标准通用标记语言)的一种应用,是一系列用作标记语言的元语言。SGML 相当复杂,实际上大多数浏览器都不完全遵循它所有的奇怪特性。在网络的实际应用中,HTML 是一个受 SGML 影响的语言。

关于 HTML 需要注意的另一点是所有的 HTML 用户代理程序(这是用来称呼那些阅读 HTML 的程序的术语,包括网络浏览器、搜索引擎、网络爬虫等)都有相当宽松的错误处理机制。许多技术上不合法的结构,像错误的嵌套标签或者不好的属性名,都是被允许的。这种错误处理在不同浏览器间是相对一致的。但是在一些边缘情况下仍然有许多不同,因为这种错误处理行为不是书面化的,也不是任何标准的一部分。所以,把编写好的 HTML 文档在不同的浏览器中验证并通过,是一个良好的做法。

HTML5 是 HTML 的最新标准。

HTML5 的提出是用来代替原来的 HTML4(XHTML)版本。

HTML5 的设计是为了使该语言支持更多的多媒体格式,可读性更强,支持更多的计算机和其他智能设备,比如,HTML5 目前支持 PC、平板电脑、智能手机、智能电视等。

提示: HTML5 与 HTML4 并无原则上的区别,HTML5 是 HTML4 的进化与强化版本,更好地支持多媒体功能、支持 PC 以外更多的设备。

2.1.4 为 HTML 做好准备

因特网上的许多页面充斥着“不规范的”HTML。

如果只是在浏览器中查看,那么下列 HTML 代码看起来会运行得很好(虽然它没有遵循 HTML 规则)。

例 2.3 一个“不规范的”HTML 代码文件。

```
<html>
<head>
  <title>This is bad HTML</title>
  <body>
  <h1>Bad HTML
  </body>
```

为了标准化 HTML 代码,必须遵循下列 4 条主要的规则。

1. HTML 元素必须正确嵌套

在 HTML 中,一些元素之间可以不正确地嵌套,如:

例 2.4 不正确的嵌套。

```
<b><i>This text is bold and italic</b></i>
```

在 HTML 中,所有元素之间必须正确地嵌套,如:

例 2.5 正确的嵌套。

```
<b><i>This text is bold and Italic</i></b>
```

注意: 关于嵌套列表的一个常见错误,就是忘记了内层列表必须在
和
标签

之间。

例 2.6 “不规范的”代码：嵌套中忘记了一个。

```
<ul>
<li>Coffee</li>
<li>Tea
  <ul>
    <li>Black tea</li>
    <li>Green tea</li>
  </ul>
<li>Milk</li>
</ul>
```

例 2.7 规范代码：加上
完成正确嵌套。

```
<ul>
<li>Coffee</li>
<li>Tea
  <ul>
    <li>Black tea</li>
    <li>Green tea</li>
  </ul>
</li>
<li>Milk</li>
</ul>
```

注意：在正确代码例子中
标签后面插入了一个
标签。

2. HTML 元素必须是闭合的

非空元素必须有一个终止标签。

例 2.8 “不规范的”代码：无终止标签导致不闭合。

```
<p>This is a paragraph
<p>This is another paragraph
```

例 2.9 规范代码：加入终止标签。

```
<p>This is a paragraph</p>
<p>This is another paragraph</p>
```

空元素也必须闭合。

例 2.10 “不规范的”代码：空元素没有闭合。

```
A break: <br>
A horizontal rule: <hr>
An image: <img src="happy.gif" alt="Happy face">
```

例 2.11 规范代码：空元素闭合。

```
An break: <br />
A horizontal rule: <hr />
An image: <img src="happy.gif" alt="Happy face" />
```

3. HTML 元素必须小写

标签名和属性必须小写。

例 2.12 “不规范的”代码：出现了大写的标签名。

```
<BODY>
<P>This is a paragraph</P>
</BODY>
```

例 2.13 规范代码：HTML 格式应该是标签名小写。

```
<body>
<p>This is a paragraph</p>
</body>
```

4. HTML 文档必须有一个根元素

所有 HTML 元素必须嵌套在<html>这个根元素内。子元素必须成对并且正确嵌套在它们的父元素中。

基本文档结构如下：

例 2.14 HTML 基本文档结构。

```
<html>
<head> …</head>

<body> … </body>
</html>
```

html 元素是 HTML 和 HTML 文档中的最外层元素。html 元素也被叫做根元素。

<html>标签告诉浏览器这是一个 HTML 文档。

head 元素是一个容器，head 内部所有的元素都被包含在这个容器内。<head>中的元素可以包括脚本、指引浏览器到指定位置寻找样式表以及提供元信息等。下列标签可以被加入头部：<base>、<link>、<meta>、<script>、<style>和<title>。<title>标签定义了文档的标题，它也是头部内唯一一个必须的元素。

body 元素包含在网络浏览器中显示的所有内容，如文本、超级链接、图像、表格、列表等。

一个仅包含最少的必须标签的简单 HTML 文档如下：

例 2.15 HTML 必须包含如下标签。

```
<html>
<head>
  <title>Title of the document</title>
</head>
<body>
The content of the document… …
</body>
</html>
```

例 2.15 的网页显示如图 2.2 所示。

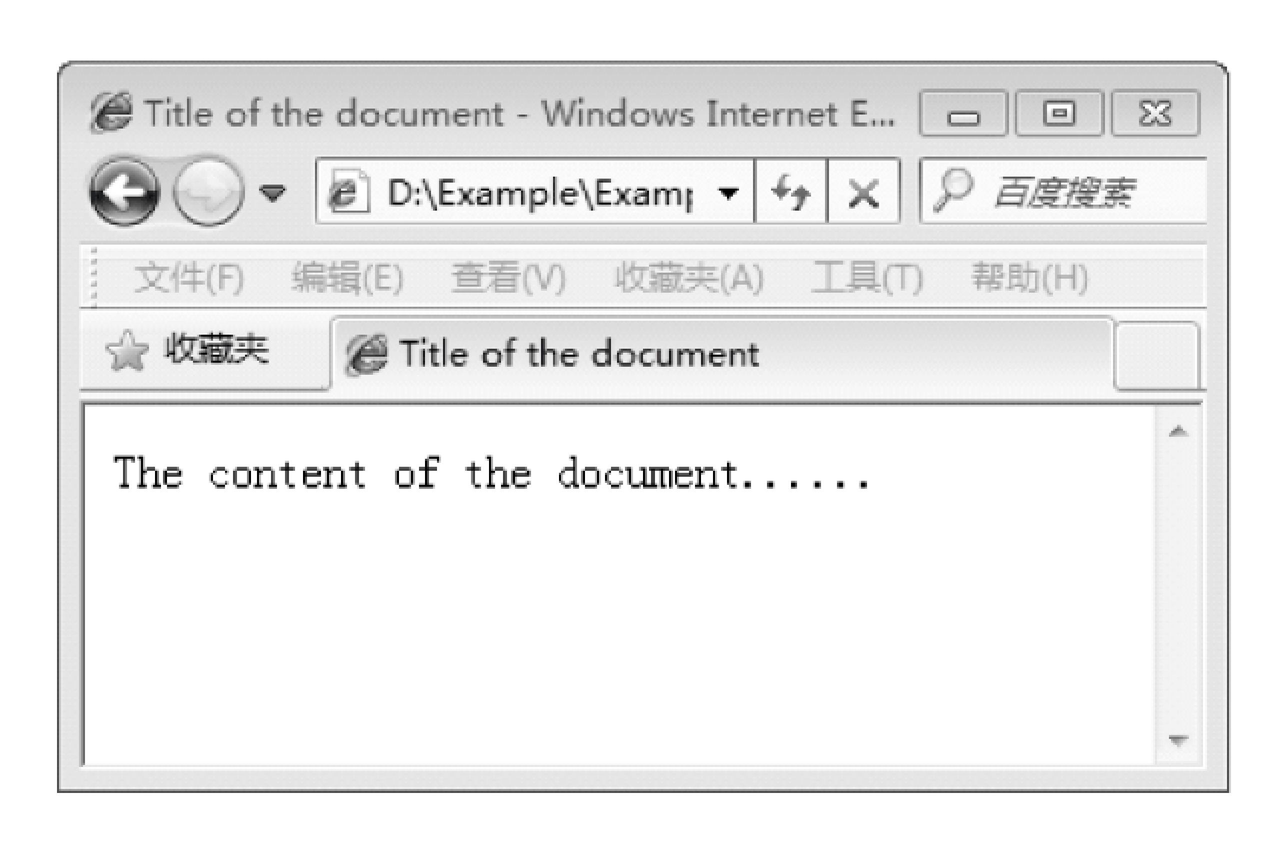


图 2.2 包含最少的必须标签的网页

更多的一些 HTML 语法规则会在相关章节介绍。

提示：有关 HTML 的规范要求还有其他内容，在今后的讲解中逐步提到。

注意即使是所谓规范的 HTML 编码，文档中的最外层标签仍然是<html>标签，而不是<HTML>，文件扩展名也仍然是 html(或 htm)。

2.2 文本元素

从本章开始学习基本的 HTML 标签和属性。

2.2.1 标题

例 2.16 演示标题标签是如何工作的。

例 2.16 标题标签。

```
<html>
<head>
  <title>Title of the document</title>
</head>
<body>
  <h1>This is heading 1</h1>
  <h2>This is heading 2</h2>
  <h3>This is heading 3</h3>
  <h4>This is heading 4</h4>
  <h5>This is heading 5</h5>
  <h6>This is heading 6</h6>
</body>
</html>
```

图 2.3 是例 2.16 在浏览器中的显示效果。

<h1>到<h6>标签用来定义 HTML 标题。

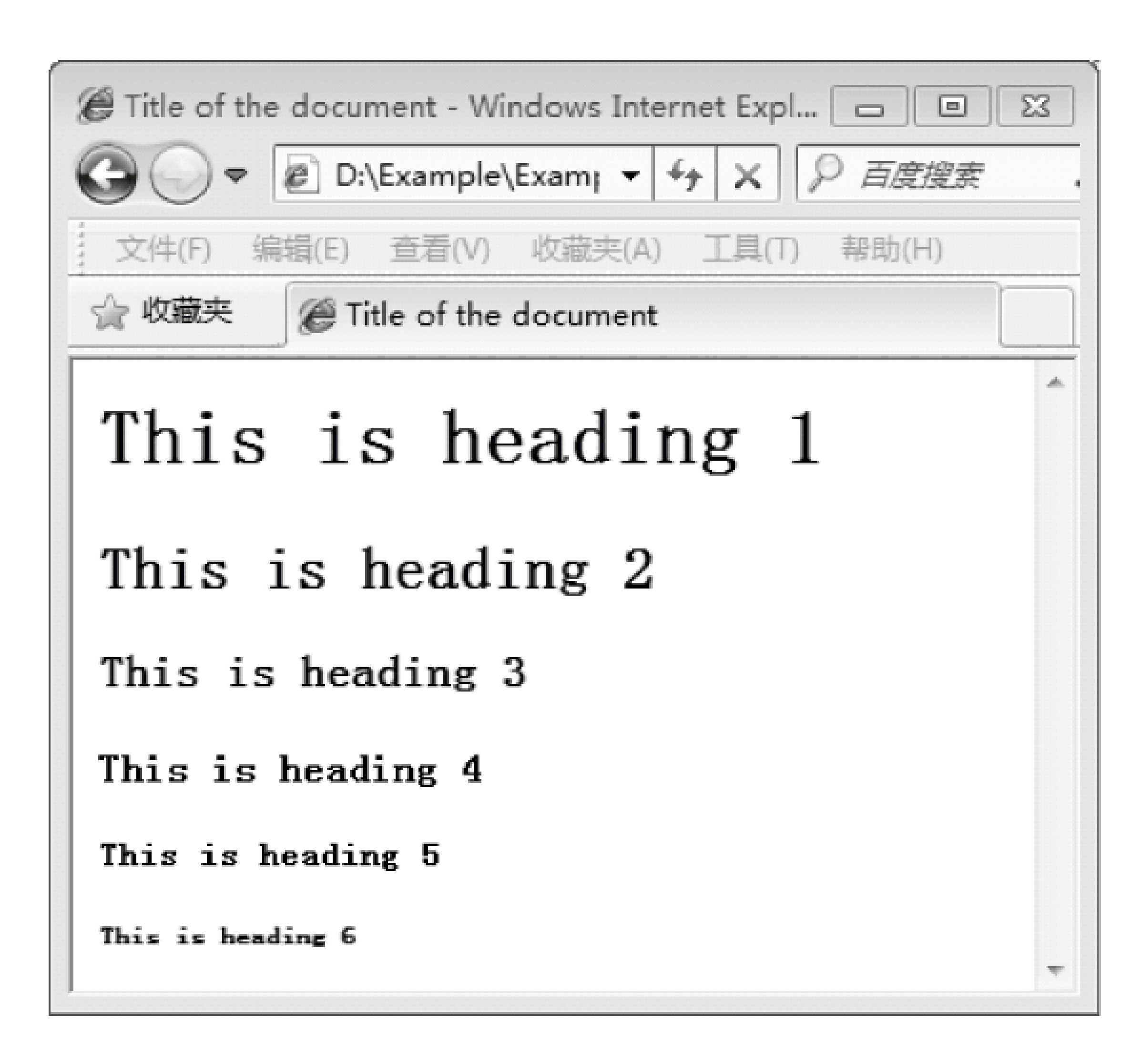


图 2.3　标题 1 到标题 6

标题(h 元素)语法:

```
<h# align="left|center|right">标题内容</h#>
```

#（值从 1 到 6)是标题的大小,<h1>定义字号最大的标题,<h6>定义字号最小的标题。

align 是一个可选属性,它指定标题的对齐方式。属性值为 left、center 和 right 其中之一,分别设定为左对齐、居中和右对齐。

属性:标签还可以添加属性,来赋予它额外的信息。属性出现在起始标签内部,它的值出现在引号内并且必须是小写。它们的格式看起来像<tag attribute="value">content…</tag>这样。一旦为标签加入属性,标签将能起更多的作用。

例 2.17　加入属性。

```
<html>
<head>
  <title>Title of the document</title>
</head>
<body>
  <h1 align="left">Heading 1 is on the left.</h1>
  <h2 align="center">Heading 2 is in the middle.</h2>
  <h3 align="right">Heading 3 is on the right.</h3>
</body>
</html>
```

例 2.17 在浏览器中显示效果如图 2.4 所示。

与例 2.2 对比,看一看例 2.18 中的元素结构。

例 2.18　属性概念。

```
<h1 align="left">Heading 1 is on the left.</h1>
```

<h1>是起始标签而</h1>是终止标签,Heading 1 is on the left. 是中间的内容。在

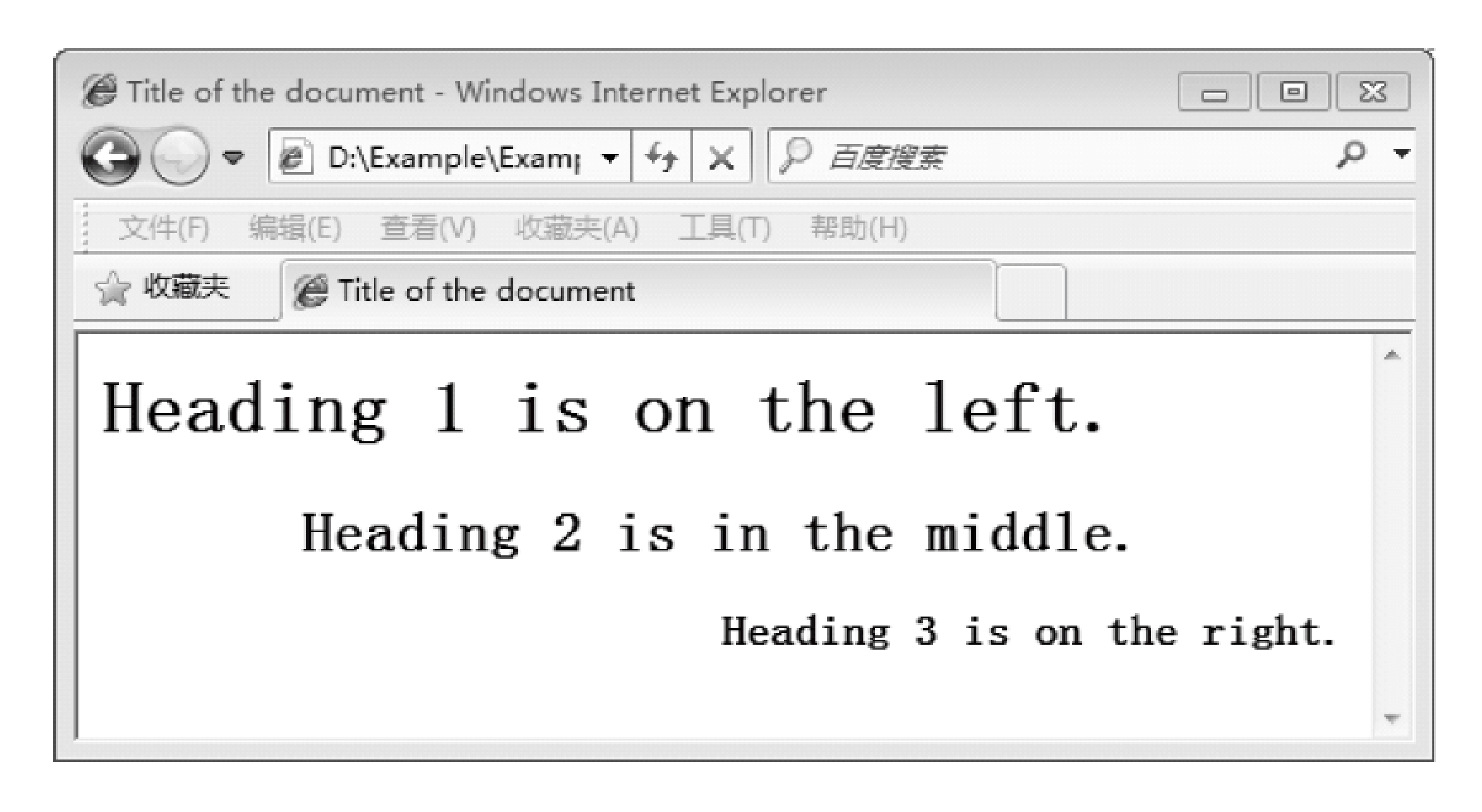


图 2.4 <h>标签的对齐属性

起始标签中加入可选属性来丰富元素。

注 1：<h#>标签和<head>标签是不一样的。

注 2：<h1>到<h6>元素的 align 属性在 HTML 4.01 中不推荐，而在 HTML 1.0 的 Strict DTD 文档声明之下是不被支持的，可用 CSS 来替代。

提示：通过标题文字标签又学习到标签内的 attribute(属性)，标签内的属性丰富了标签的应用。要注意文中对属性的语法要求，而对于使用中文的读者，还要注意在编辑代码时使用英文的双引号如："left"，而不要使用中文的双引号："left"，否则浏览器无法识别。

还有，align 属性按严格的 HTML 标准已经不再使用，现在用的是更规范的 CSS 办法来实现同样效果，第 3 章将详细介绍 CSS。

2.2.2 文档标题

<title>标签定义文档的标题，即在标题栏内显示的内容。

title 元素：

(1) 定义浏览器标题栏中的标题。

(2) 提供页面在收藏夹中标题。

(3) 在搜索引擎搜索结果中显示标题。

每个 HTML 文档在头部必须有一个 title 元素。注意<title>元素位于<head>部分，不是位于<body>中。

网站设计者应该用 title 元素来给文档内容定义一个标识。由于用户通常脱离背景或上下文来查阅文档，所以作者应该提供富有语境背景的标题。因此，作者应该提供一个诸如 Introduction to Nordic Walking("北欧健走介绍")的标题，而不是一个仅仅像 Introduction 这样的一个没有提供太多语境背景的标题。

文档标题(title 元素)语法：

```
<title>title's text</title>
```

将例 2.17 中<title>元素内容替换为"align attribute of <h> tag"，如例 2.19 所示。

例 2.19 显示文档标题。

```
<html>
<head>
  <title>align attribute of <h> tag</title>
</head>
<body>
  <h1 align="left">Heading 1 is on the left.</h1>
  <h2 align="center">Heading 2 is in the middle.</h2>
  <h3 align="right">Heading 3 is on the right.</h3>
</body>
</html>
```

浏览器显示例 2.19 的效果如图 2.5 所示。

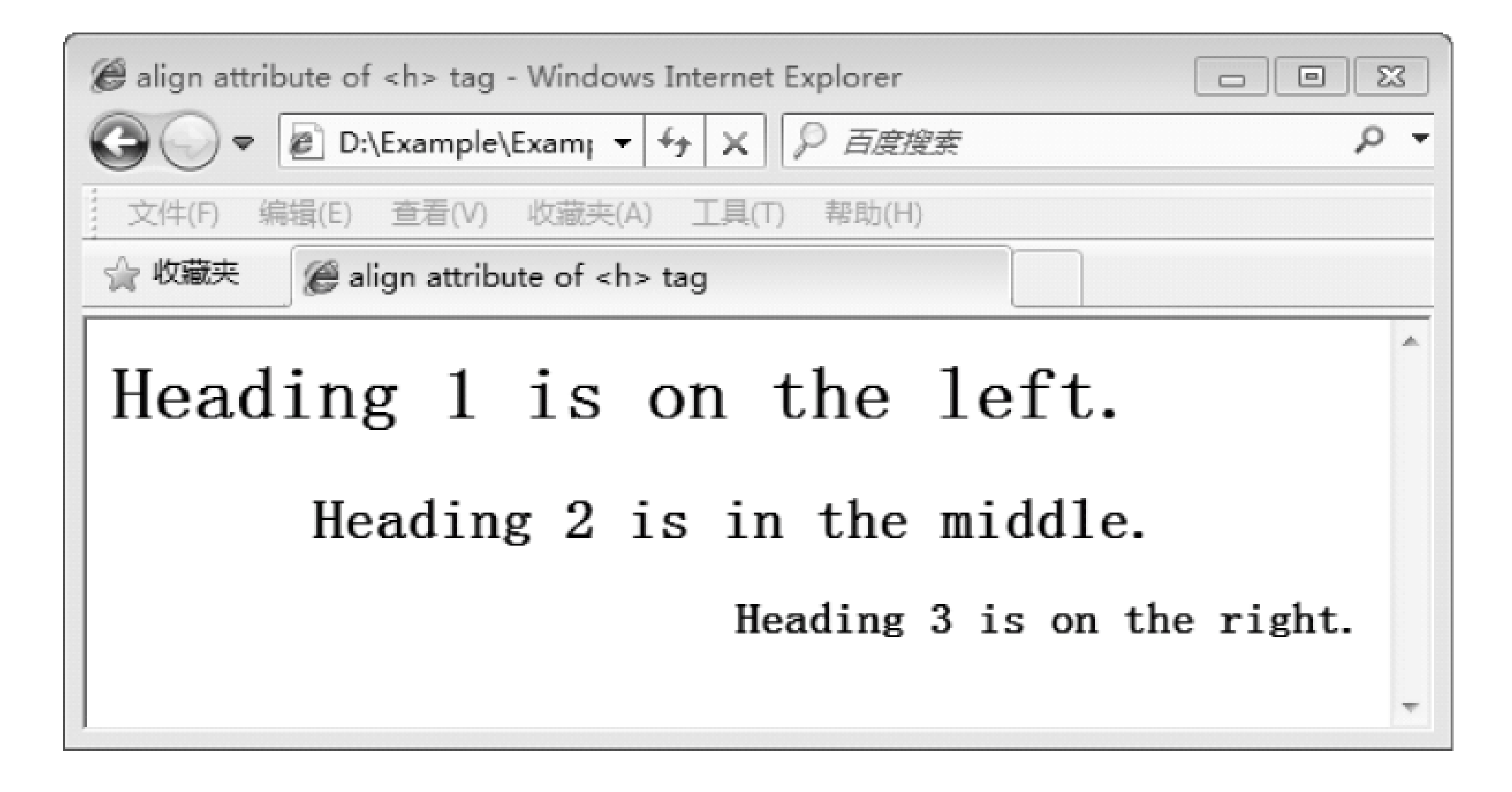


图 2.5 每个 HTML 文档必须有一个 title 元素

提示：本章中所讲解的标签绝大多数是出现在<body>部分，<title>标签是个例外，它出现在<head>部分，这点要注意。本章只要不特别指明，标签都是在<body>内使用。

注意<title>标签与<h>标签的不同。

2.2.3 段落

段落(Paragraph)使用 p 元素完成。p 元素会在它自身前后自动创建一些空间。这些空间由浏览器自动应用，也可以在样式表(CSS)中指定。

段落(Paragraph 元素)语法：

```
<p align="left|center|right"> text </p>
```

例 2.20 段落元素应用。

```
<html>
<head>
  <title>Show paragraphs</title>
</head>
<body>
  <h1>This is heading 1</h1>
  <p>This is paragraph 1.</p>
```

```
  <p>This is paragraph 2.</p>
  <h3>This is heading 3</h3>
</body>
</html>
```

例 2.20 在浏览器中显示效果如图 2.6 所示。

看不到特别的地方吗？那么把图 2.6 中浏览器窗口内的所有内容复制下来，然后粘贴到记事本中。

现在该知道将网页的内容拷贝粘贴到记事本或者 Word 程序中的时候，为什么在段落之间出现这么多额外的空行了吧。

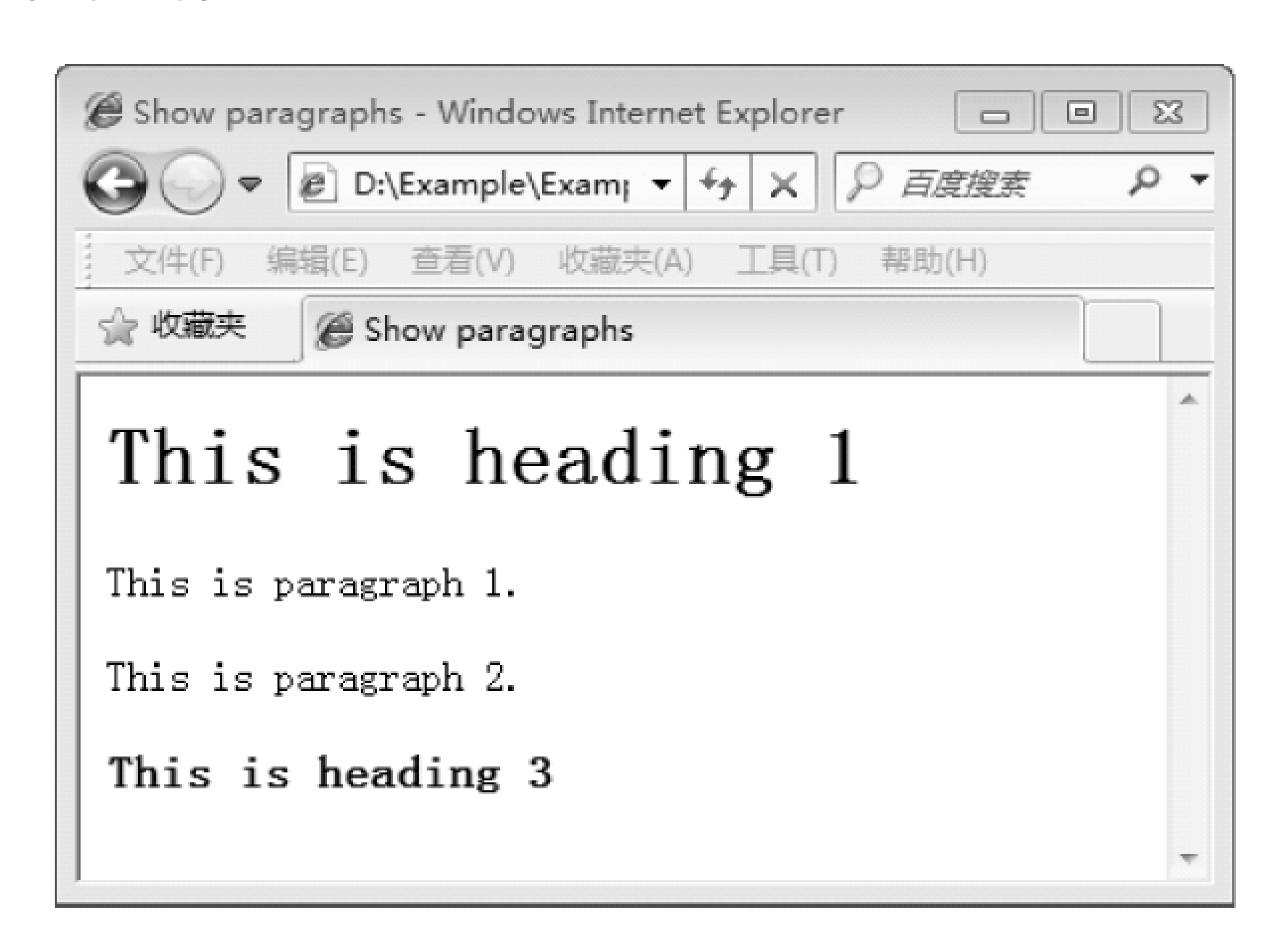


图 2.6 段落元素

段落的一些其他默认行为。

例 2.21 段落的默认行为。

```
<html>
<head>
  <title>Some other default behaviors of paragraphs</title>
</head>
<body>
  <p>
  This paragraph
  contains a lot of lines
  in the source code,
  but the browser
  ignores it.
  </p>
  <p>
  This paragraph
  contains a lot of spaces
  in the source code,
  but the browser
  ignores it.
  </p>
  <p>
```

```
      The number of lines in a paragraph depends on the size of your browser window. If you
resize the browser window, the number of lines in this paragraph will change.
  </p>
</body>
</html>
```

例 2.21 在浏览器中显示效果如图 2.7 所示。

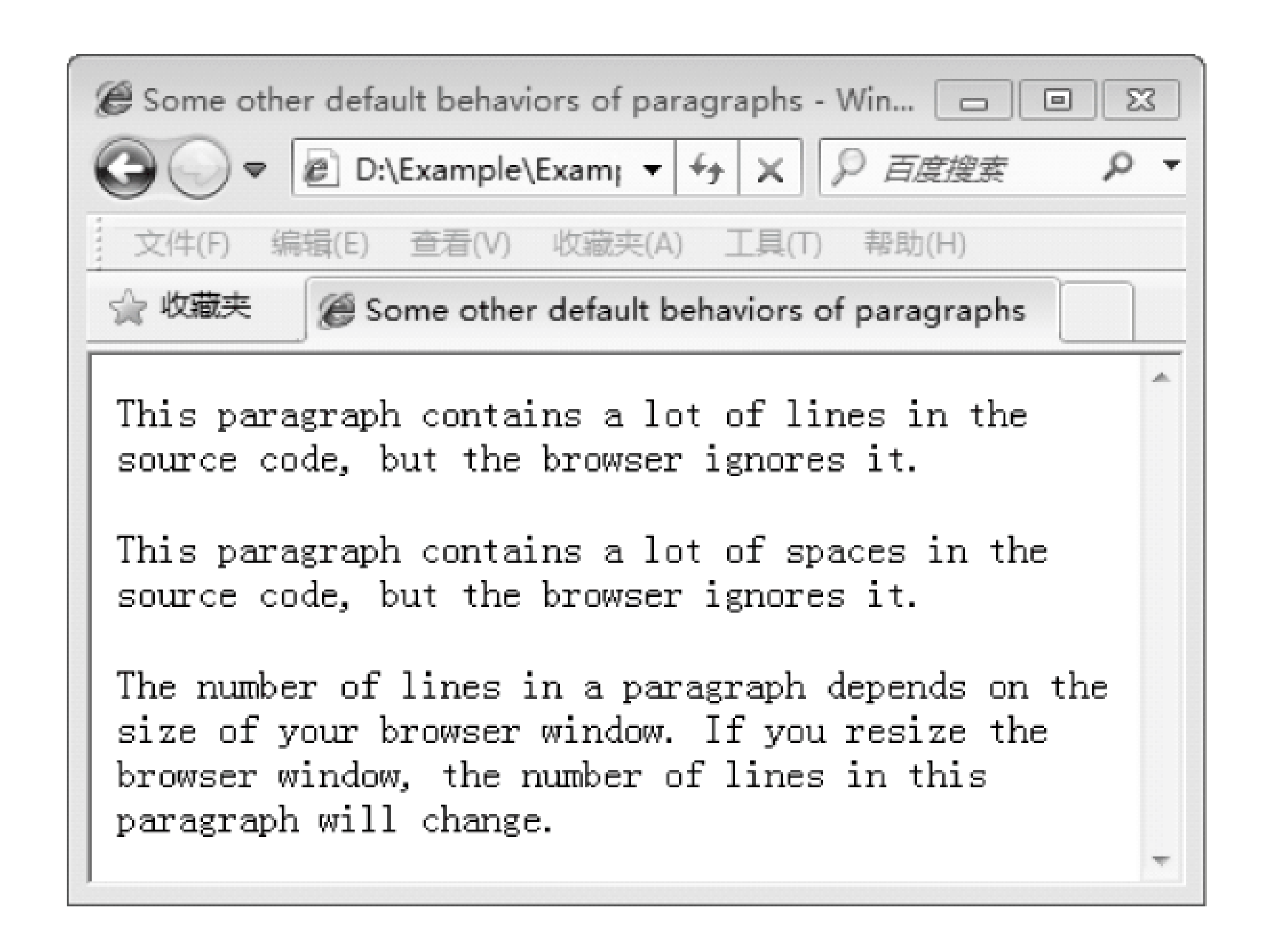


图 2.7 段落的一些其他默认行为

从现在开始,本教材将带领读者一步一步利用已经讲授的知识,为一个健身俱乐部——Walker Club——建设一个网站。

项目 2.1 Walker Club 主页(index.html 版本 1)。

```
<html>
<head>
  <title>沃克俱乐部欢迎您</title>
</head>

<body>
  <h1 align="center">咱们走着瞧——欢迎光临沃克俱乐部</h1>
  <h3>走路,简单有效的健身之道</h3>
    <p>走路是人类本能的身体活动,人类身体结构就是为步行设计的。问题是,随着汽车等代步工
具的普及,人们"好逸恶劳"的条件和理由越来越充足,以致基本的走路功能都有些退化了。根据美国
疾病控制与预防中心的研究,有高达 75% 的美国人每天运动量不到 30 分钟,大部分人几乎成天坐在
椅子上,是不折不扣的"沙发土豆"。反观我们中国的现实情形,随着经济的发展,汽车的大量使用及
其他因素,导致缺乏锻炼、体形肥胖的人越来越多。
  </p>
    <p>走路健身又健脑。我国自古就有"走为百练之祖"的健身经验谈,其中传统医学认为,双脚是
人体的健康之根。走路刺激脚底穴位,能舒筋通络,活血顺气,强身健体。同时,现代运动医学研究也
证实:走路时,骨骼、肌肉、韧带、神经末梢都要参加运动,从而促进血液循环,调节大脑皮层的活动功
能,促使身体各种激素分泌,使人心情愉悦。走路防病并延缓衰老。最新的医学研究表明,一周健步
走 7 小时以上,可以降低 20% 乳腺癌、30% 心脏病和 50% 糖尿病的罹患率,而中老年人每天散步 2.4
公里以上,心脏病发作率将降低 50%。
  </p>
</body>
</html>
```

项目 2.1 在浏览器中显示效果如图 2.8 所示。



图 2.8　为 Walker Club 创建主页

提示：从这里开始，教材中出现了一个 Project（项目），这个项目建立一个虚拟的俱乐部网站。它不同于为讲解某一个标签或属性而做的例子，它是对前边所学内容的综合应用到一个网站建设中，随着学习内容的不断深入，这个网站项目会不断应用新的学习内容，从而使该网站逐步变得更加丰富充实。

2.2.4　换行

标签会插入一个单行的换行。

元素是一个没有任何内容的元素，因为它仅仅意味着一个换行，没有别的任何意思。
不是唯一一个这样的元素；还有其他类似的元素，它们有一个共同的名字：空元素。在 HTML 中
标签没有终止标签，但在 HTML 中要求
标签必须被正确的关闭，就像这样：
。

从 p 标签可以看出浏览器忽略 p 元素内的回车和空格键创建出来的换行和空格。
标签被用来强制换行。

换行（br 元素）语法：

```
Text <br />
```

将例 2.22 插入<body>元素中，理解一下
标签的功能。

例 2.22　换行效果。

```
This text is on the first line.
<br />
And this text is on the second line.
```

单就换行效果本身来说，

```
<p>text</p>
```

等于

```
text<br /><br />
```

例 2.23 换行与强制不换行。

```
<html>
<head>
  <title>br & nobr tags</title>
</head>
<body>
  This text is on the first line.
  <br />
  And this text is on the second line.
  <br />
      <nobr> Change the width of browser's window and make it smaller than this line. What happens?
  </nobr>
</body>
</html>
```

例 2.23 在浏览器中显示的效果如图 2.9 所示。

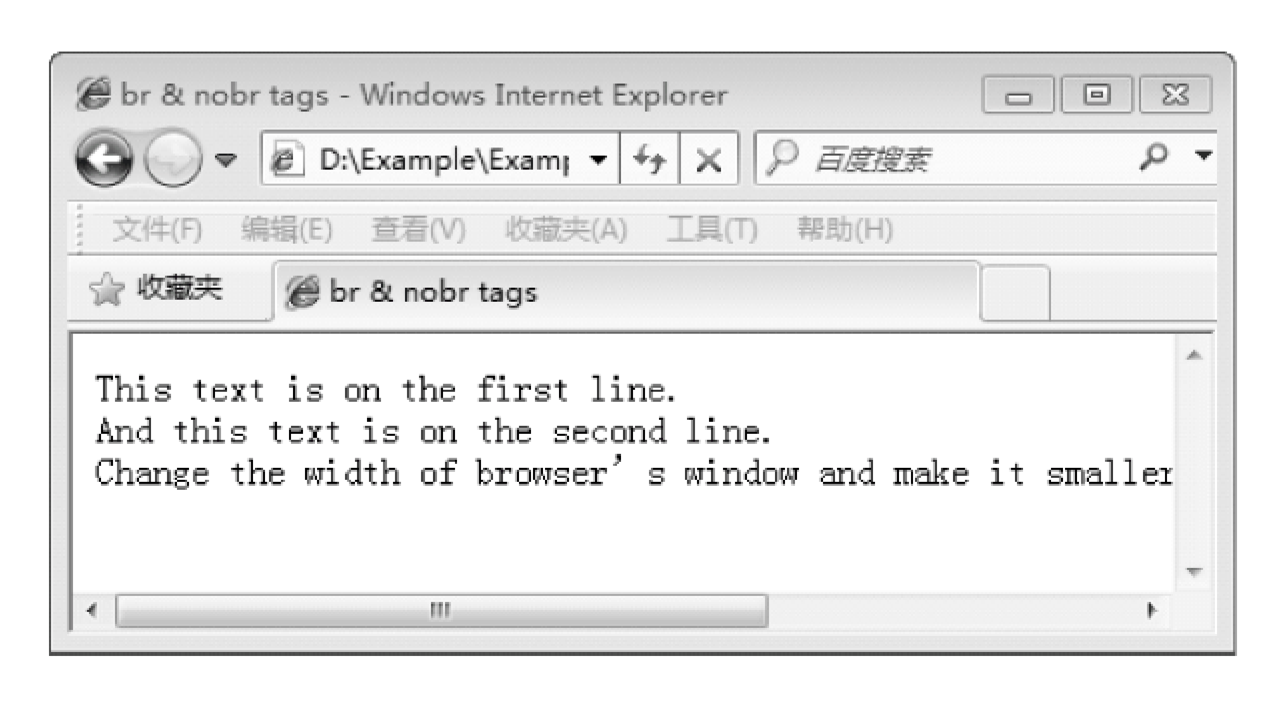


图 2.9 br 和 nobr 标签

<nobr>标签用来禁止这个标签间的文本的任何换行。

提示：从<p>标签举例可以知道，在网页制作中无法使用回车或者空格来完成强制换行或增加字符间空格的功能。强制换行使用
标签，强制不换行则使用标签<nobr>，这个标签对换行可能引起误读的公式、一行长数字等尤其有用。但注意这个标签是 HTML 的产物，在 HTML 设计方法中已经不再使用（虽然浏览器还可以识别），将来学习 CSS 时，有更为规范的办法实现该功能。

如前所述，HTML 只识别文档中的一个空格，多余空格一律忽略，如果要加入多个空格，可以用“ ”（non-breaking space）插入一个空格，即非换行空格，反复使用它可以在文本中加入任意多个空格。

2.2.5 水平线

<hr>会在 HTML 页面中创建一条水平线。

hr 元素是一个空元素，用来分隔 HTML 页面中的内容。

水平线语法：

```
<hr align="left|center|right" size="pixels" width="pixels or %" color="color or color code" />
```

size：水平线的高度(粗细)。

width：水平线的宽度。

color：水平线的颜色，使用颜色名称或是 RGB 颜色代码。一些常见的颜色和颜色代码如表 2.1 所示。

表 2.1 颜色——RGB 代码对照表

Color	Color Code
black	#000000
blue	#0000ff
brown	#a52a2a
cyan	#00ffff
gray	#808080
green	#008000
ivory	#fffff0
orange	#ffa500
pink	#ffc0cb
red	#ff0000
white	#ffffff
yellow	#ffff00

提示：<hr>是又一个空标签。需要说明的是上述<hr>标签内出现的属性都是HTML 时代的产物，按严格的 HTML 标准已经不再使用。为什么本教材还要讲解规范的HTML 不再使用的属性呢？这里有两点考虑：其一是让读者可以看懂当前仍然大量存在的 HTML 代码；其二是当第 3 章讲到 CSS 的相关内容时，读者通过对比可以更好地理解为什么 CSS 的办法要优于现在这种在标签内大量加入属性的办法，对 HTML + CSS 这一当前高效的静态网页开发有更深刻的理解。

2.2.6 注释

注释标签(<! --……-->)用来在源代码中插入一段注释。注释会被浏览器所忽略。用注释来解释编写的代码，可以在以后再次编辑源代码时帮助理解。

注释中也可以存放一些特定程序信息，在这种情况下它们对用户将是不可见的，但是对程序仍然是可见的。一个很好的实际应用是把脚本或者样式元素的文本放在注释中，以防止早期不支持脚本或样式的浏览器把它们当成普通文本显示出来。

注释语法：

```
<! -- comments -->
```

例 2.24 使用注释使代码更具可读性。

```
<html>
<head>
```

```
  <title>horizontal rules & comments</title>
</head>
<body>
  <h3>This text is on the first line.</h3>
  <hr /><! -- default attribute of hr element -->
  <h3 align="center">And this text is on the second line.</h3>
  <! -- absolute width of hr element -->
  <hr align="center" size="12" width="500" color="blue"/>
  <! -- <h3>This third line is commented.</h3> -->
</body>
</html>
```

在浏览器中的显示效果如图 2.10 所示。

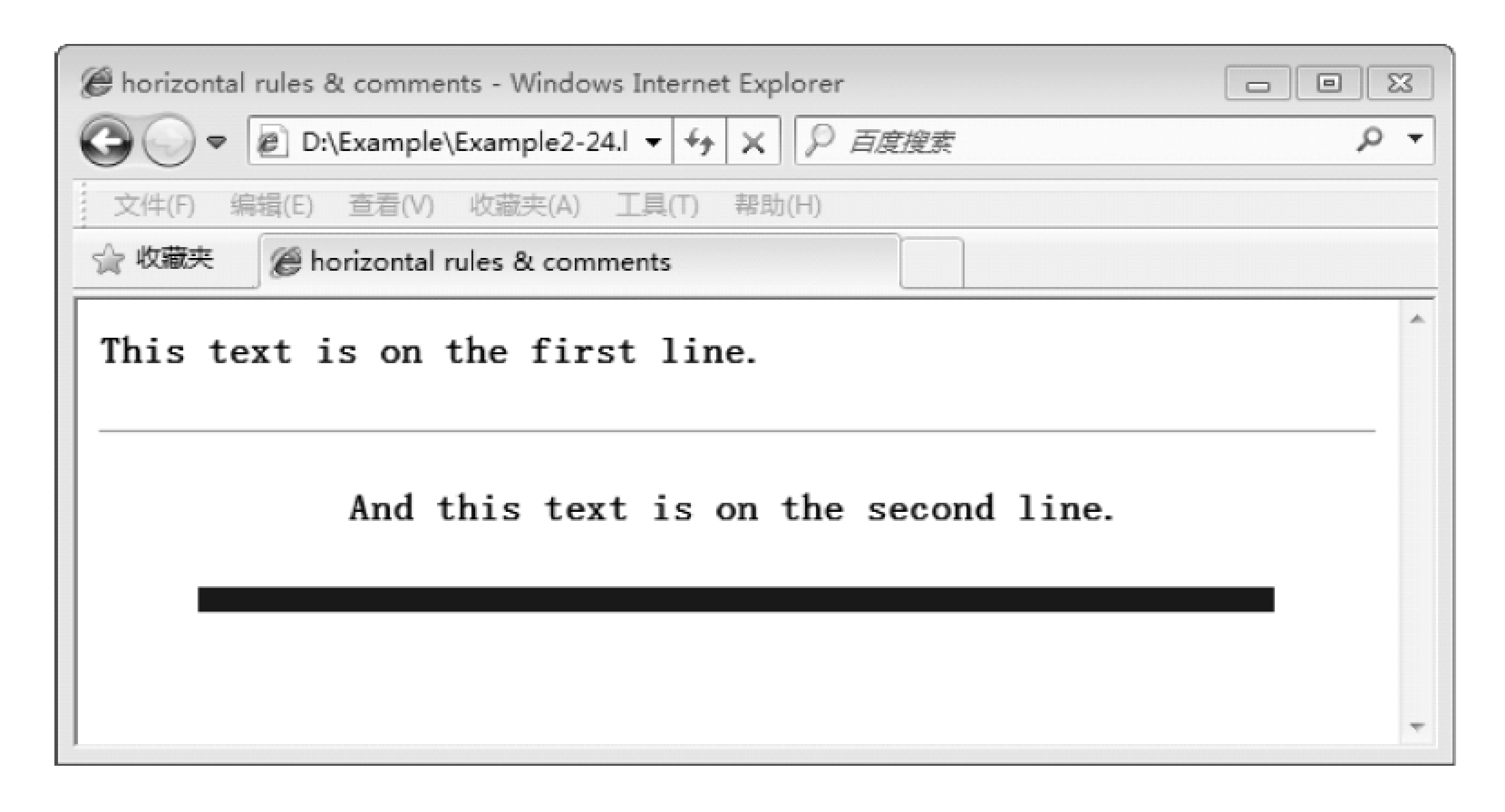


图 2.10 水平线和注释

提示：与许多计算机语言一样，HTML 文档也提供注释功能。浏览器会忽略此标签中的文字(可以是许多行)而不显示在窗口中。注释标签可以为文档加上说明，也可以为了调试方便而暂时让部分内容不显示。

2.2.7 节元素<div>

<div>标签在 HTML 文档中定义了一个独立的分区或者部分。

<div>标签通常用来将块级元素分组归类以便用样式表 CSS 来格式化它们。

节元素(<div>)语法：

```
<div>content (text, image or table)</div>
```

例 2.25 使用节元素。

```
<html>
<head>
  <title>Use division</title>
</head>
<body>
      <div style="color:green"><! -- define a section with all contents in green -->
        <h3>This is a header</h3>
        <p>This is a paragraph.</p>
```

```
  </div>
</body>
</html>
```

项目 2.2 Walker Club 主页(index.html 版本 2)。

```
<html>
<head>
  <title>沃克俱乐部欢迎您</title>
</head>
<body>
      <h1 align="center">咱们走着瞧——欢迎光临沃克俱乐部</h1><!-- 显示主标题 -->
  <hr width="80%" /><!-- 标题下的横线设定为窗口宽度的 80% -->
  <h3>走路,简单有效的健身之道</h3>
  <div style="text-indent:1.5em"><!-- 设定首行缩进 -->
      <p>走路是人类本能的身体活动,人类身体结构就是为步行设计的。问题是,随着汽车等代步工具的普及,人们"好逸恶劳"的条件和理由越来越充足,以致基本的走路功能都有些退化了。根据美国疾病控制与预防中心的研究,有高达 75% 的美国人每天运动量不到 30 分钟,大部分人几乎成天坐在椅子上,是不折不扣的"沙发土豆"。反观我们中国的现实情形,随着经济的发展,汽车的大量使用及其他因素,导致缺乏锻炼、体形肥胖的人越来越多。</p>
      <p>走路健身又健脑。我国自古就有"走为百练之祖"的健身经验谈,其中传统医学认为,双脚是人体的健康之根。走路刺激脚底穴位,能舒筋通络,活血顺气,强身健体。同时,现代运动医学研究也证实:走路时,骨骼、肌肉、韧带、神经末梢都要参加运动,从而促进血液循环,调节大脑皮层的活动功能,促使身体各种激素分泌,使人心情愉悦。走路防病并延缓衰老。最新的医学研究表明,一周健步走 7 小时以上,可以降低 20% 乳腺癌、30% 心脏病和 50% 糖尿病的罹患率,而中老年人每天散步 2.4 公里以上,心脏病发作率将降低 50%。</p>
  </div>
  <hr />
  <div>
  <!-- 这部分用来定义版权声明及联系方式 -->
  </div>
</body>
</html>
```

项目 2.2 在浏览器中的显示效果如图 2.11 所示。

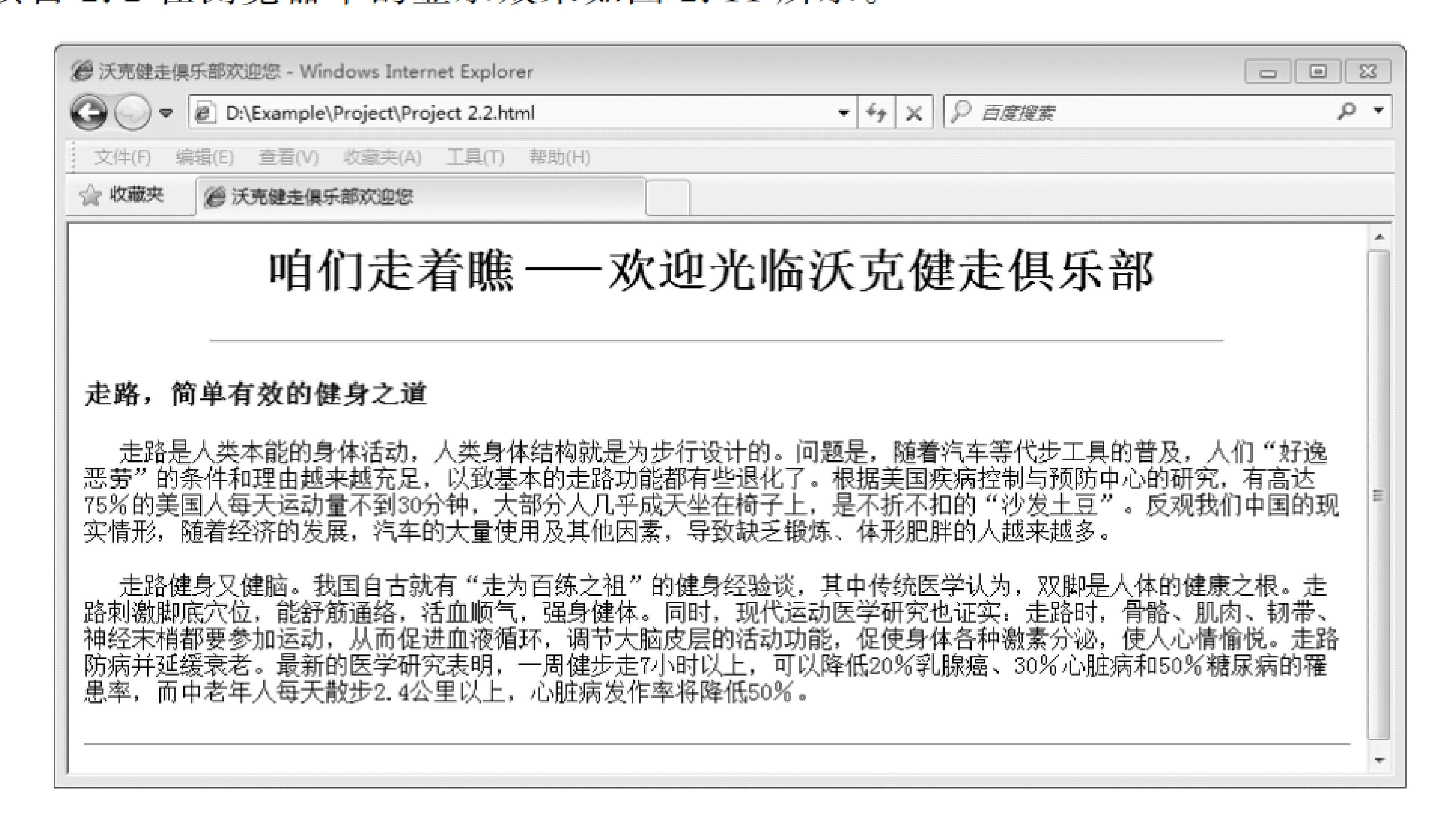


图 2.11 改良的 Walker Club 主页

div 元素经常用于和 CSS 一起来布局网页。

提示：文本标签的学习到这里就结束了。这些只是最基本的标签，由于篇幅的关系，教材不可能把所有的标签与属性都涉及到，完整的标签及定义请看书后附录。在例 2.25 和项目 2.2 中的<div>标签都出现了以前没有讲到的属性，这些将引导更深入地学习 Web 应用技术。

2.3 超级链接

在网络术语中，一个超级链接是网络中一个资源的标识(一个地址)。超级链接可以指向网络上的任何资源：一个 HTML 页面、一幅图片、一个声音文件、一个电影等等。锚是一个用来定义目标在文档内部的超级链接目标的术语。但 HTML 锚元素<a>既可被用来定义超级链接也可以被用来定义页内锚链接。

a 元素通常指一个链接或超级链接。

2.3.1 <a>标签

例 2.26 用<a>标签定义了一个链接到例 2.25 的链接。

例 2.26 最简单的超链接。

```
<html>
<body>
  <a href="Example2-25.html">Link to Example 2.25</a>
</body>
</html>
```

在浏览器中的显示效果如图 2.12 所示。

Link to Example 2.25

图 2.12 用<a>标签来链接另一个文档

超链接(a 元素)语法：

```
<a href= "url">Link text</a>
```

<a>元素最重要的属性是 href 属性，它定义了链接的“目的地址”。起始标签包含了关于链接的属性。而元素内容(链接文本)定义了显示的部分。

注意：元素内容不一定必须是文本，图片或者任何其他 HTML 元素上都可以创建链接。

在浏览器中，默认情况下，未被访问的链接是蓝色带有下划线，已被访问的链接为紫色带下划线，活动链接为红色带下划线。

提示：超链接是 Web 的最重要特性，完成这个功能的是<a>标签。<a>标签取自于英文 anchor(锚点)，指的是本文档中另外一个需要链接的位置。事实上，在<a>标签中设

定不同的属性，不仅可以完成到本文档中的链接，也可以完成到其他文档、其他网站的链接，完成自动调用电子邮件客户端及文件下载等。本节将详细讲解这些不同属性的应用。

2.3.2 路径和目录

路径是用目录名和文件名，指定文件系统中的一个独一无二的位置。路径的层次遵循目录的树状层次结构，常见的格式是用斜杠（“/”）和反斜杠（“\”）分隔开目录名来代表目录。路径在计算机科学中应用广泛，通常在现代的操作系统中表示目录/文件关系，同时它更是统一资源定位符（URL）的必要组成。

系统既可以用绝对路径也可以用相对路径。一个完全的路径即绝对路径，不管当前的工作目录或链接的路径改变与否，都指向文件系统中的同一个位置。它通常直接引用到根目录。相对路径是一个相对于当前用户或应用程序的工作目录的路径，也就是只从当前用户或应用程序的工作目录开始引用，并不给出完整的路径。

例 2.27 绝对路径。

```
D:\WebApp\Project\index.html
http://www.walkerclub.net/index.html
```

例 2.28 相对路径。

```
Project\index.html
index.html
```

就像上面例子中看到的那样，绝对路径包含了全部路径，而相对路径只包含其中的一部分。

绝对路径的典型特点就是通过域名来指向位于另一域名的网页元素。例如，链接到 Google 网站——需要做的就是像例 2.29 那样把域名包含到 URL 中。

例 2.29 超链接到另外一个网站要使用绝对路径。

```
<a href = "http://www.google.com"> Search in Google </a>
```

如果引用的一个网页元素是在相同的域名下，那么相对路径是个好的选择。相对路径根据链接所在的页面不同而改变。用相对路径创建链接有 4 条规则：

(1) 链接到当前目录内的文件。

```
<a href = "filename.html"> Link text </a>
```

(2) 链接到子目录内的文件。

```
<a href = "subdirectory/filename.html"> Link text </a>
```

(3) 链接到父目录内的文件。

```
<a href = "../filename.html"> Link text </a>
```

“../”表示返回到父目录。

(4) 链接到同级目录内的文件。

```
<a href = "../sibling directory/filename.html"> Link text </a>
```

“../sibling directory/”表示先返回到父目录，然后进入到同级目录。

提示：在网页设计中，一定要清楚绝对路径（Absolute Path）和相对路径（Relative Path）的概念。绝对路径只是在引用本网站之外其他网站时，把其他网站的绝对路径即完整URL放入href属性值中，使用相对简单。而在创建自己的网站时，大量使用的是相对路径，这样才能使得整个网站项目被放置到不同的硬盘空间或者域名时，可以保证相互链接的完整性与一致性。

熟练掌握相对路径是用好计算机语言的前提，在HTML设计中，它是用好<a>标签的前提，所以要认真学习掌握本节链接到不同关系目录的文件时，<a>标签中相对路径引用的不同写法。下一节将讲解具体的例子。

2.3.3 组织网站目录结构

在开始创建更多的HTML页面之前，需要把文件组织一下。目前，项目目录（文件夹）中只有一个index.html文件。如果把网页、图像和其他的资源组织进一组不同的目录（文件夹）中，将更便于管理。现在给予Walker Club一个如图2.13和图2.14的有意义的目录结构。

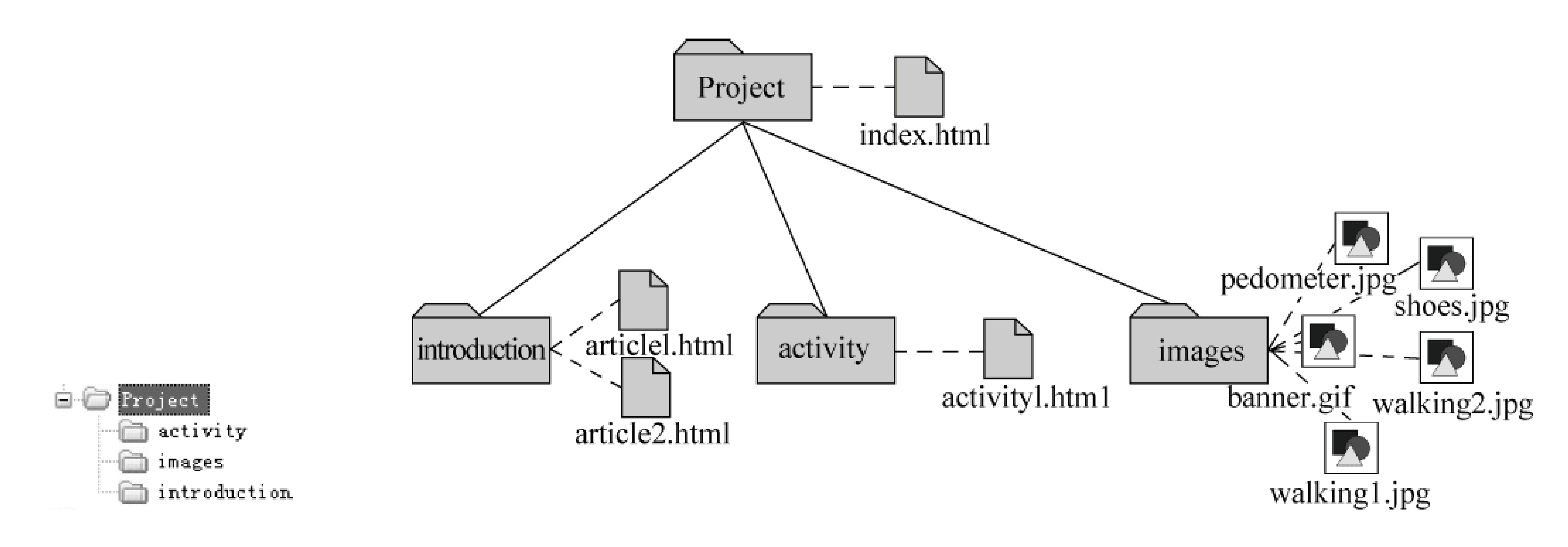


图 2.13 项目目录结构

图 2.14 Walker Club网站的目录结构

下面的4个例子根据前一节学习的相对路径的4条规则，介绍了如何参照图2.14中的关系创建超级链接。

例 2.30 从article1.html中链接到article2.html。

```
<a href = "article2.html"> article2 </a>
```

例 2.31 从index.html中链接到article2.html。

```
<a href = "introduction/article2.html"> article2 </a>
```

例 2.32 从activity1.html中链接到index.html。

```
<a href = "../index.html"> Home </a>
```

例 2.33 从article1.html中链接到activity1.html。

```
<a href = "../activity/activity1.html"> activity1 </a>
```

提示：在设计网站时，各种文件组织到目录（文件夹）中的方法不是唯一的，只要便于设

计者或者设计团队管理维护即可。一般原则来说，可以按网站栏目或者版块进行文件分类，或者按照文件被访问安全权限进行分类，建议主目录下有一个独立的images目录，便于进行图片的管理；网站目录层次不要太深，建议不超过3层，便于设计者维护管理；目录名及文件名不要使用中文，因为Web是全球网络，中文目录名或文件名可能会对网址的正确显示造成困难。

这里给出了Walker Club网站的文件及目录（文件夹）结构，同时举例说明通过相对路径链接到目标文件时href属性的写法。今后要讲到的<img>标签在引用图片时，将会用到同样的相对路径引用办法。

2.3.4 target属性

<a>标签的target属性定义了被链接的文档将在哪里打开。下面的代码（例2.26的修改版）将在一个新浏览器窗口打开被链接的文档。

例2.34 在一个新浏览器窗口打开被链接的文档。

```
<html>
<body>
  <a href="Example2-25.html" target="_blank">Link to Example 2.25</a>
</body>
</html>
```

比较例2.26和它的修改版本例2.34。单击例2.26页面中的Link to Example 2.25时，浏览器在当前窗口显示例2.25（它替换了例2.26的页面）。但是单击例2.34页面上的Link to Example 2.25，例2.25会在一个新浏览器窗口中打开（例2.34仍然留在原来窗口中）。这样的好处是显而易见的——访客在新浏览器窗口中打开链接，因而他们不会被迫或意外离开最初的网站。

提示：<a>标签的target属性有4个可选值来确定链接在什么地方打开，其中在新的浏览器窗口打开（target="_blank"）是最常见的应用。这种效果可以在CSS中实现，HTML建议使用CSS方法实现。

2.3.5 链接到页面内部的一个位置

当<a>元素指向一个资源时，它在术语中叫做链接或超链接。但是当<a>元素定义文档内部的一个地址时，我们叫它页内链接。前面已经讨论了超链接，本节讨论页内链接。

当使用了<a>元素中的name属性，<a>元素就在HTML文档内部定义了一个书签标记。

书签标记语法：

```
<a name="label">Any content</a>
```

链向一个书签标记语法：

```
<a href="#label">Link text</a>
```

href属性中的“#”定义了一个链向书签标记的链接。

例 2.35 创建页内链接。

```
<html>
<head>
  <title>Link to a bookmark</title>
</head>
<body>
  <p><a href="#C6">See Chapter 6</a></p>
  <h2>Chapter 1</h2>
  <p>This chapter discusses …</p>
  <br /><br /><br />
  <h2>Chapter 2</h2>
  <p>This chapter discusses …</p>
  <br /><br /><br />
  <h2>Chapter 3</h2>
  <p>This chapter discusses …</p>
  <br /><br /><br />
  <h2>Chapter 4</h2>
  <p>This chapter discusses …</p>
  <br /><br /><br />
  <h2>Chapter 5</h2>
  <p>This chapter discusses …</p>
  <br /><br /><br />
  <h2><a name="C6">Chapter 6</a></h2>
  <p>This chapter discusses …</p>
  <br /><br /><br />
  <h2>Chapter 7</h2>
  <p>This chapter discusses …</p>
  <br /><br /><br />
  <h2>Chapter 8</h2>
  <p>This chapter discusses …</p>
</body>
</html>
```

例 2.35 在浏览器中的效果如图 2.15 所示。

书签标记在任何方式下都不会显示，它们对浏览者是不可见的。页内链接通常被用于在篇幅很大的文档的开头创建“目录”——文档中的每一章被给予一个书签标记，而文档的顶部放置链向这些书签标记的页内链接。

提示：<a>标签原意锚点(Anchor)指的就是网页内的跳转，也形象地称做书签(Bookmark)。从用法上来看，网页内跳转需要对<a>标签成对地使用 href 和 name 两个属性，name 属性来定义书签位置，href 属性来链接到这个属性指定的书签位置。

由于人的阅读习惯，在网页设计时，网页宽度不能超过显示屏，但长度相对可以任意长，以方便人们使用鼠标滚轮上下翻屏。在这种网页设计中，灵活运用锚点的书签功能，网页内的上下跳转可以使网站用户迅速定位要阅读的内容。

至此，读者已经学习了<a>标签的两个最重要的超链接用法：资源链接(链接到其他网页)和锚点或书签(网页内跳转)。

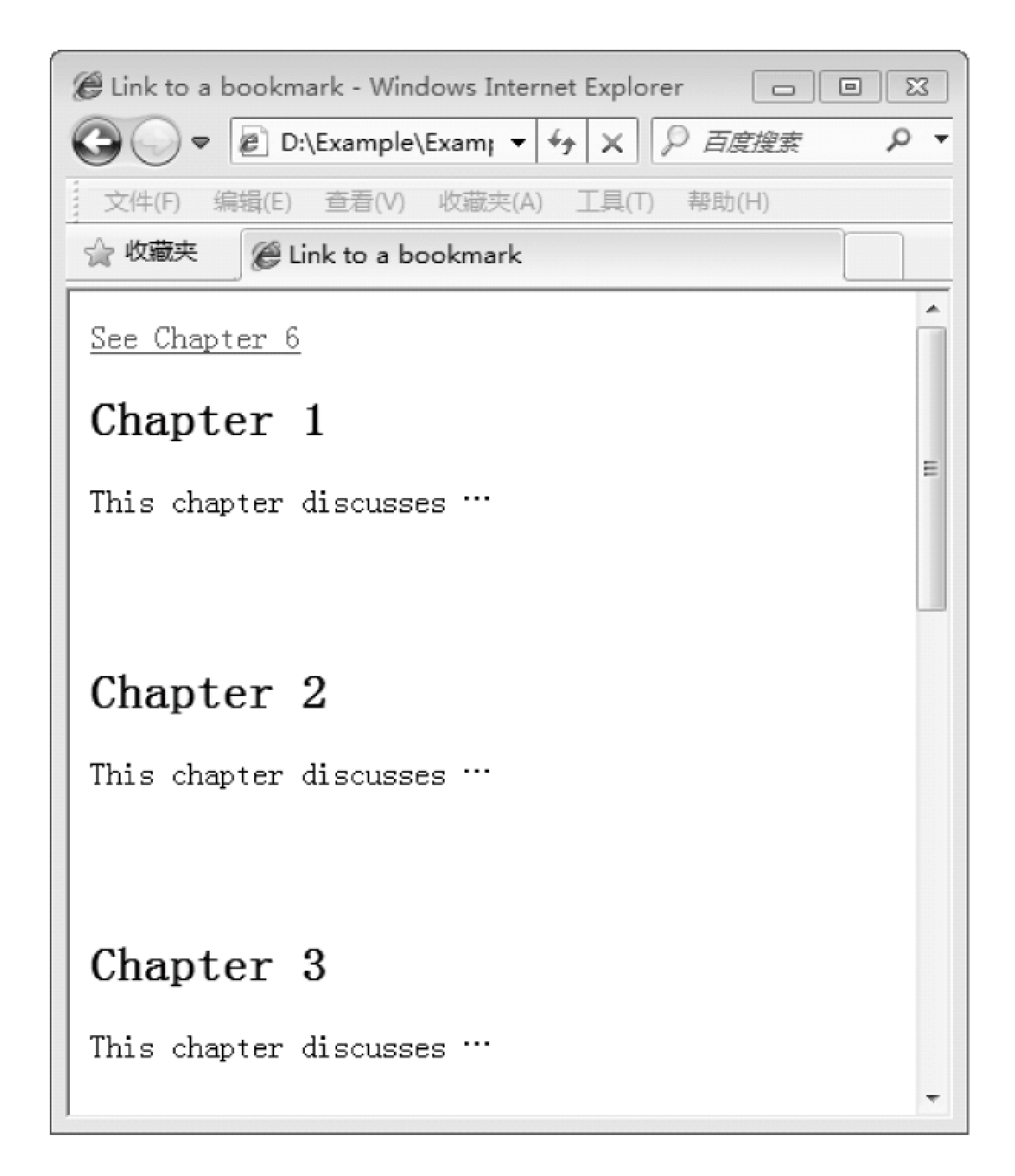


图 2.15　链向网页内部的书签

2.3.6　邮件链接

基本上每个网站都会有一两个 mailto 链接，以便让人们能从网页中直接发送电子邮件。

一个标准的邮件链接是这样的：

```
<a href = "mailto:someone@site.com"> Link text </a>
```

例 2.36　创建邮件链接。

```
<html>
<body>

      <a href = "mailto:webmaster@walkerclub.net"> webmaster@walkerclub.net (Click to
Email Me)</a>
</body>
</html>
```

例 2.36 在浏览器中显示效果如图 2.16 所示。

webmaster@walkerclub.net (Click to Email Me)

图 2.16　链向 E-mail 的链接

单击例 2.16 中的链接，就像任何邮件链接的预期效果那样——超级链接文本会激活一个已经填好收件人地址的新邮件信息窗口（只对安装的电子邮件客户端，如 Outlook

Express、Foxmail 等有效),非常方便。

提示:对指向电子邮件的链接,单击后即可打开电子邮件客户端(如 Outlook Express、Foxmail 等)并自动填写好收件人,但前提是必须使用这些电邮客户端进行电子邮件的收发(一般需要支持 POP3 协议)。如果使用基于网页的邮箱 Webmail,则本功能无法使用。所以为方便以上两种电子邮件用户发出邮件,在链接的热点文本上最好也给出实际的电邮地址,以便 Webmail 用户使用拷贝粘贴的办法把电邮地址填写到收件人处。

“mailto:”还可以实现同时给多人发邮件以及事先设定好标题(Subject)及内容(Body)等。

2.3.7 创建下载链接

有时网站需要允许访客下载一些内容到他们自己的个人电脑内。如一些大型文档和 PDF 文件、视频或者歌曲。但是大多数文件会被自动地在浏览器窗口中打开而不是被下载保存。

这里介绍两种解决办法——一种简单方法和一种稍复杂的方法。第一种需要网站用户来做绝大多数工作,而第二种要求网站制作人员为要下载的文件做一些工作。

方法一:

(1) 把要被下载的文件上传到网站服务器的指定目录。

(2) 编辑网页并添加一个标准的锚链接到文档中。

```
<a href = "large_document.pdf"> Download the large document </a>
```

(3) 在链接旁添加文字告诉访客他们需要右击链接来下载它。

(4) 访客右击链接并选择“目标另存为”来把文档保存到电脑上。

如果访客忽略了这个步骤,需要用第二种方法把文件转换为会被绝大多数浏览器自动下载的文件类型:一个 rar 文件或压缩文件。

方法二:

(1) 用 WinRAR 压缩软件把一个需要被下载的文件压缩为 rar 文件。

(2) 把 rar 文件上传到网站服务器。

(3) 编辑页面,为这个 rar 文件添加一个标准锚链接。

```
<a href = "large_document.rar"> Download the large document </a>
```

要点:

大多数操作系统有一些内置的压缩软件。如果没有,在搜索引擎中查找 WinRAR 来下载安装。

这种技术可以用于图片、电影、音乐和文档以及 PDF 文件。任何可以被压缩为 rar 文件的东西都可以发布到网站上以供下载。

提示:至此,<a>标签的重要属性及应用学习完毕。虽然只是一个<a>标签,但和不同的属性、不同路径及不同的文件格式,可以组合成非常灵活的应用。在今后更多其他标签的学习中,读者会接触到超级链接新的应用。

下面将前面所学的超链接应用到 Walker Club 网站上。

2.3.8 超链接综合应用

项目 2.3 沃克俱乐部主页(index.html version 3)。

```
<html>
  <head>
    <title>沃克俱乐部欢迎您</title>
  </head>
  <body>
    <h1 align="center">咱们走着瞧——欢迎光临沃克俱乐部</h1><!-- 显示主标题 -->
    <hr width="80%" /><!-- 标题下的横线设定为窗口宽度的 80% -->
    <div style="text-align:center">
    <!-- 使用 target 属性,使 article2.html 在新窗口打开 -->
      <p>
        <a href="introduction/article1.html" target="_blank">健走的方法</a> 
  
        <a href="introduction/article2.html" target="_blank">健走装备</a>  
 
        <a href="activity/activity1.html" target="_blank">俱乐部活动</a>
      </p>
    </div>
    <h3>走路,简单有效的健身之道</h3>
    <div style="text-indent:1.5em"><!-- 设定首行缩进 -->
      <p>走路是人类本能的身体活动,人类身体结构就是为步行设计的。问题是,随着汽车等代
步工具的普及,人们“好逸恶劳”的条件和理由越来越充足,以致基本的走路功能都有些退化了。根据
美国疾病控制与预防中心的研究,有高达 75% 的美国人每天运动量不到 30 分钟,大部分人几乎成天
坐在椅子上,是不折不扣的“沙发土豆”。反观我们中国的现实情形,随着经济的发展,汽车的大量使
用及其他因素,导致缺乏锻炼、体形肥胖的人越来越多。</p>
      <p>走路健身又健脑。我国自古就有“走为百练之祖”的健身经验谈,其中传统医学认为,双
脚是人体的健康之根。走路刺激脚底穴位,能舒筋通络,活血顺气,强身健体。同时,现代运动医学研
究也证实:走路时,骨骼、肌肉、韧带、神经末梢都要参加运动,从而促进血液循环,调节大脑皮层的活
动功能,促使身体各种激素分泌,使人心情愉悦。走路防病并延缓衰老。最新的医学研究表明,一周
健步走 7 小时以上,可以降低 20% 乳腺癌、30% 心脏病和 50% 糖尿病的罹患率,而中老年人每天散步
2.4 公里以上,心脏病发作率将降低 50%。</p>
  </div>
    <div style="text-align:center"><!-- 网页尾部设为文本居中对齐 -->
      <hr />
      <p><a href="mailto:webmaster@walkerclub.net">和我联系 Email: webmaster@
walkerclub.net</a>
      </p>
    </div>
  </body>
</html>
```

项目 2.3 在浏览器中的显示效果如图 2.17 所示。



图 2.17 改进后的沃克俱乐部主页(version 3)

项目 2.4 article1 的 HTML 文档(article1.html)。

```
<html>
  <head>
    <title>沃克俱乐部 - 健走的方法</title>
  </head>
  <body>
  <! -- 将本网页首部书签设为 top，本网页其他部分可以用“回首部”来回到首部 -->
    <p><a href = "../index.html" name = "top">主页</a>   
    <! -- 使用 target 属性,使 article2.html 在新窗口打开 -->
      <a href = "article2.html" target = "_blank">健走装备</a>
    </p>
    <hr />
    <h3>健走的方法</h3>
    <p><a href = "#jzyl">健走要领</a></p><! -- 设定网页内跳转 -->
    <div style = "text - indent: 1.5em">
      <p>健走即俗称的快走。健走如果步伐适当,其速度也不亚于慢跑。在现代的竞走运动中,一些运动员能达到 3 分 30 秒/公里的速度,一个优秀的竞走运动员,20 公里只需要一个半小时就能完成。这个速度对于跑友来说,都已经相当快了。与跑步相比,快走通常更持久,而且,如上所述也能达到很大的强度,足够锻炼心肺功能,同时还能发展腿部、腰部和背部肌肉的力量。</p>
      <p>在我国古代的军队中,就有一种急行军比赛,要求士兵快走完成比赛,而不是跑,当时的距离是 90 公里,要求在 8 个小时之内完成。可见,在古代就有用快走锻炼身体的传统。</p>
      <p>健走比慢跑好的地方也有几点,一是不容易对足部、脚踝和膝盖造成伤害,二是没有特殊的要求,只要不妨碍迈步,普通服装也可以。</p>
      <p>另外,健走和跑步一样,形式也是多种多样的,可以在海边走,也可以在山上走,还可以在室内走,但健走也有一个标准,那就是时
 速不能低于 6 公里。</p>
```

```
        <p><a name="jzyl">健走的动作要领如下：   </a><a href="#top">回首部</a></p>
        <ol><!-- 设定排序列表 ordered list -->
          <li>身体前倾 3～5 度，抬头，肩和背自然下垂，挺胸，收腹。</li>
          <li>两臂必须配合摆动，手臂摆动的高度不得高于胸。</li>
          <li>脚跟着地要柔和，快速滚动到全脚掌着地。</li>
          <li>两脚落地时以内侧为准踩成一条直线。</li>
          <li>不要出现双脚都腾空的现象。</li>
          <li>健走要求走路跨大步、速度敏捷、双臂摆动、抬头挺胸，比慢跑安全，也比散步有效。</li>
        </ol>
      </div>
      <div style="text-align:center"><!-- 网页尾部设为文本居中对齐 -->
        <hr />
        <p><a href="mailto:webmaster@walkerclub.net">和我联系 Email: webmaster@walkerclub.net</a></p>
      </div>
    </body>
  </html>
```

项目 2.4 在浏览器中的显示效果如图 2.18 所示。



图 2.18 article1.html 页面效果

项目 2.5 article2 的 HTML 文档(article2.html)。

```
<html>
  <head>
    <title>沃克俱乐部 - 健走装备</title>
  </head>
```

```
<body>
<!-- 将本网页首部书签设为 top, 本网页其他部分可以用"回首部"来回到首部 -->
  <p><a href = "../index.html" name = "top">主页</a>   
  <!-- 使用 target 属性,使 article2.html 在新窗口打开 -->
    <a href = "article1.html" target = "_blank">健走的方法</a>
  </p>
  <hr />
  <h3>健走装备</h3>
  <div style = "text-indent: 1.5em">
    <p>健走因其老少皆宜,锻炼时对身体损伤小,对预防心血管病及减肥都是益处多多,正在
成为一种不亚于慢跑的时尚运动。健走毕竟不同于平常的步行,是一种有一定强度的健身运动,所
以运动装备(主要是鞋)是需要考虑的重要方面。</p>
    <p>对于健走运动鞋的选择可谓是有许多不同的说法,本文为原创,来提出自己的看法。</p>
    <p>网上的关于健走鞋选择的文章相当多,但绝大多数相互拷贝粘贴,总结出来也就如下三
种是典型说法:</p>
    <ol><!-- 设定排序列表 ordered list -->
      <li>慢跑鞋是一个较好的选择;</li>
      <li>慢跑鞋不好,应该选择健步鞋;</li>
      <li>选择平底、软底鞋,比如休闲鞋就是较好的选择。</li>
    </ol>
    <p>先来看一下健走运动中脚的运动方式,在这一点上大家并没有异议: 脚跟先落地,然后
迅速滚动到前脚掌,最后脚趾用力蹬地离开。</p>
    <p>健走和普通走路类似,只是最后脚趾用力蹬地离开这一环节是多数普通步行者所没有
做到的,仔细看的话,健步是更快更有弹性的步行。</p>
    <p>知道健步运动的原理,再来分析对鞋的要求: 因为脚跟先着地,所以对鞋的后跟要求就
要有较好的减震(或叫避震)性能,所以鞋后跟应该较厚,鞋后跟较厚可以完成减震的要求,现在所说
的气垫运动鞋,多数也是放在脚后跟上。脚前掌位置应该较薄(相对于鞋后跟),便于脚前掌联合脚趾
发力向前有蹬的动作; 鞋的最前部应该上翘,因为脚趾蹬地时,脚后跟及前掌已经都离地了,所以鞋
这样的设计保证蹬地动作不受鞋前部的阻碍。</p>
    <p>综上所述,当前市面上销售的慢跑鞋已经满足了这样的要求,所以慢跑鞋是一个不错的
选择。那为什么还有健走用慢跑鞋不好或者用平底鞋等都看来都很有道理的说法呢?</p>
    <p>来分析第 2 条: 慢跑鞋不好,应该选择健走鞋。这种说法从后半段来看是一个好主意,
但毕竟现在极少有见到运动鞋厂家生产专门的健走鞋,所以健走爱好者只有等待。本说法的前半段
说慢跑鞋不好是说慢跑与健走要求不同,说慢跑是前脚掌先落地,所以慢跑鞋是前脚掌减震性较强,
而鞋后跟减震性弱。而实际情况是: 现在市面上的慢跑鞋恰恰是后跟有减震(多数气垫都在后跟),
脚前掌没有减震,原因也简单: 大众平时休闲、旅游、跑步都用慢跑鞋,最关键的是,普通人即使在慢
跑锻炼时,多数也是后跟先着地或者是后跟与脚掌同时着地的,很少有像专业长跑运动员那样脚前
掌着地,后跟根本不着地(专业长跑运动员也不用平时市面上的慢跑鞋了,他们用的可是专业的跑
鞋),所以,第 2 种说法是对当前慢跑鞋的结构明白不多,对普通老百姓慢跑的动作不了解所致。</p>
    <p>来分析第 3 条: 选择平底、软底鞋,休闲鞋就是较好的选择。本说法没什么大的问题,
平时大家都是这样穿休闲鞋的。但注意健走是至少要每小时 6 千米以上的速度,一次至少 30 分钟以
上的运动时间,这样的考虑,普通平底鞋对脚后跟没有减震,脚受到的冲击就较大,时间长会对脚有
坏处,所以,本来健走是为健康,结果时间长倒使身体某个部分出问题,这是不划算的。还有一点,就
是平底鞋一般前部也不上翘,尤其是现在比较流行的板鞋,这对长时间走路是不好的。板鞋是用来
进行滑板运动用的,平底,鞋前部也是平的,便于整个鞋子紧贴滑板并控制滑板,这种鞋看起来也美
观,鞋上可以加上各种各样的图案,功能上却不强,进行滑板运动时合适,但它却不适合大运动量健
走。</p>
    <p>最后说一下,篮球鞋和网球鞋也不适合健走,它们是为了篮球和网球这两项运动特有的
迅速起动、制动、转向等这些动作设计的,穿这些鞋健走,反倒使脚不舒服。</p>
    <p>大家都来健走吧!!!</p>
    <p>祝大家身体健康!!!</p>
```

```
            <p>总结说来：健走，目前慢跑鞋是最好的选择。   <a href = "#top">回
首部</a></p>
          </div>
          <div style = "text - align:center"><! -- 网页尾部设为文本居中对齐 -->
            <hr />
            <p><a href = "mailto:webmaster@walkerclub.net">和我联系 Email: webmaster@
walkerclub.net</a></p>
          </div>
        </body>
      </html>
```

项目 2.5 在浏览器中显示效果如图 2.19 所示。

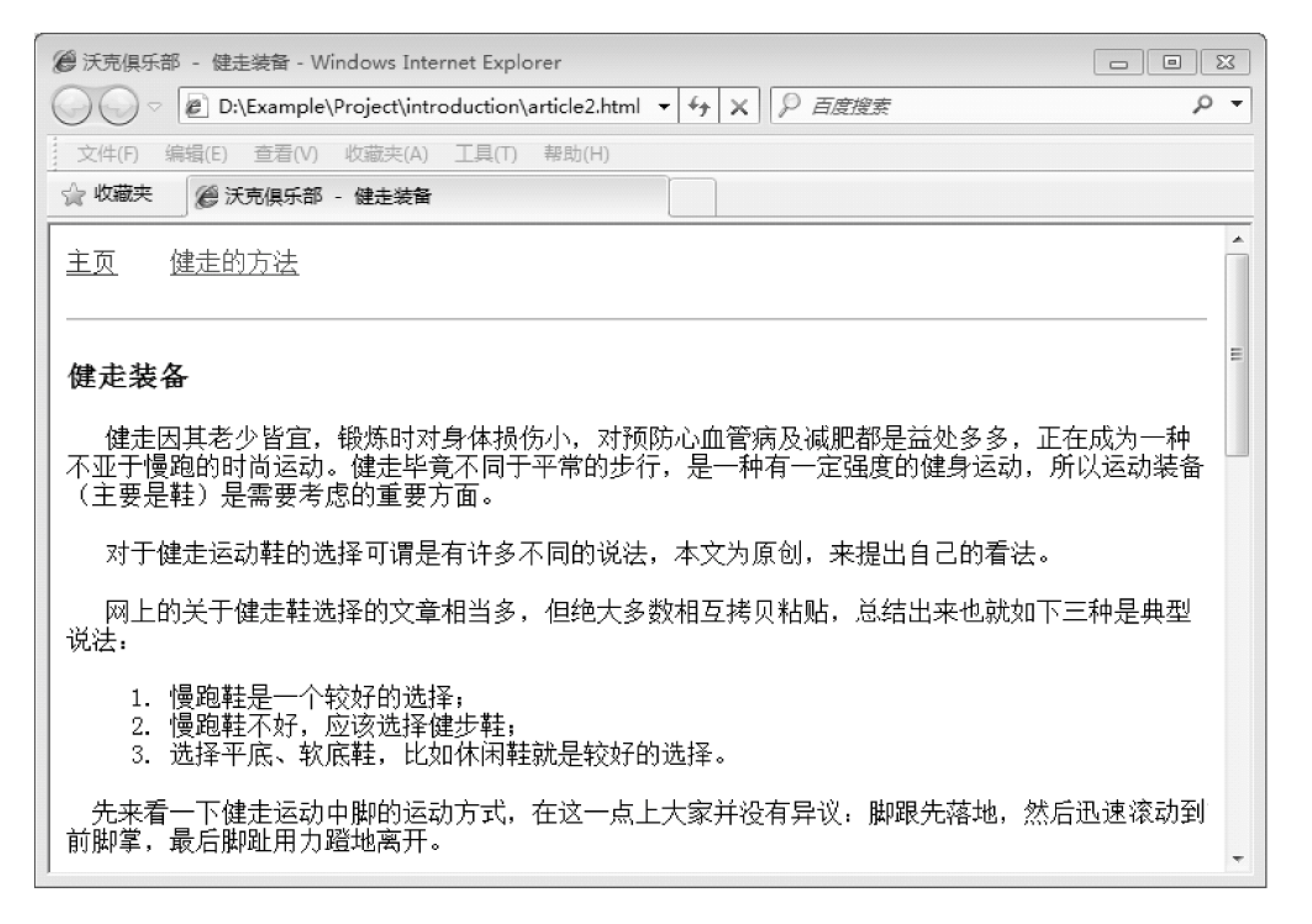


图 2.19　article2.html 页面效果

项目 2.6　activity1 的 HTML 文档(activity1.html)。

```
  <html>
    <head>
      <title>沃克俱乐部 - 俱乐部活动</title>
    </head>
    <body>
      <! -- 将本网页首部书签设为 top，本网页其他部分可以用"回首部"来回到首部 -->
      <p><a href = "../index.html" name = "top">主页</a>   
        <! -- 使用 target 属性，使 article1.html 和 article2.html 在新窗口打开 -->
        <a href = "../introduction/article1.html" target = "_blank">健走的方法</a> 
  
        <a href = "../introduction/article2.html" target = "_blank">健走装备</a>
      </p>
      <hr />
```

```
        <h3>常规活动</h3>
        <p><a href="#tbhd">特别活动</a></p><!-- 设定网页内跳转 -->
        <div style="text-indent: 1.5em">
          <p>常规活动为每周一、三、五、六进行。具体时间安排及路线安排如下：</p>
          <ul><!-- 设定非排序列表 unordered list -->
            <li>时间安排</li>
            <ul>
              <li>每周一、三、五</li>
              <ul>
                <li>6、7、8 月：6:30pm—8:00pm</li>
                <li>3、4、5、9、10、11 月：6:00pm—7:30pm</li>
                <li>12、1、2 月：5:30pm—7:00pm</li>
              </ul>
              <li>每周六</li>
              <ul>
                <li>全年：3:00pm—5:00pm</li>
              </ul>
            </ul><br />
            <li>路线安排</li>
            <ul>
              <li>每周一、三、五：从北三环与渠西路交叉口出发，沿东风渠，到经三路与东风路交
叉口折回，到北三环与渠东路交叉口结束。全长约 7 千米。</li>
              <li>每周六：从北三环与渠西路交叉口出发，顺北三环向东，到森林公园，森林公园绕
行并折回出发点，全长约 12 千米。</li>
            </ul><br />
            <li>注意事项</li>
            <ul>
              <li>自愿参加，准时出发。</li>
              <li>请穿运动鞋，携带揩汗的毛巾及足量的水，夏天注意遮阳，冬天注意防寒。</li>
              <li>如遇极端天气无法进行活动，俱乐部不再特别通知。</li>
            </ul>
          </ul>
        </div>
        <h3><a name="tbhd">特别活动安排：</a></h3>
        <div style="text-indent: 1.5em">

            <ul>
            <li>特别活动一般安排在清明、五一、国庆等假日进行，时间为一天，路线为市区近郊，健
走路程为 25 至 30 千米。具体的时间安排及路线请会员注意网站通知。   <a href
="#top">回首部</a></li>
          </ul>
        </div>
          <div style="text-align:center"><!-- 网页尾部设为文本居中对齐 -->
          <hr />
          <p><a href="mailto:webmaster@walkerclub.net">和我联系 Email: webmaster@
walkerclub.net</a></p>
        </div>
      </body>
    </html>
```

项目 2.6 在浏览器中显示效果如图 2.20 所示。

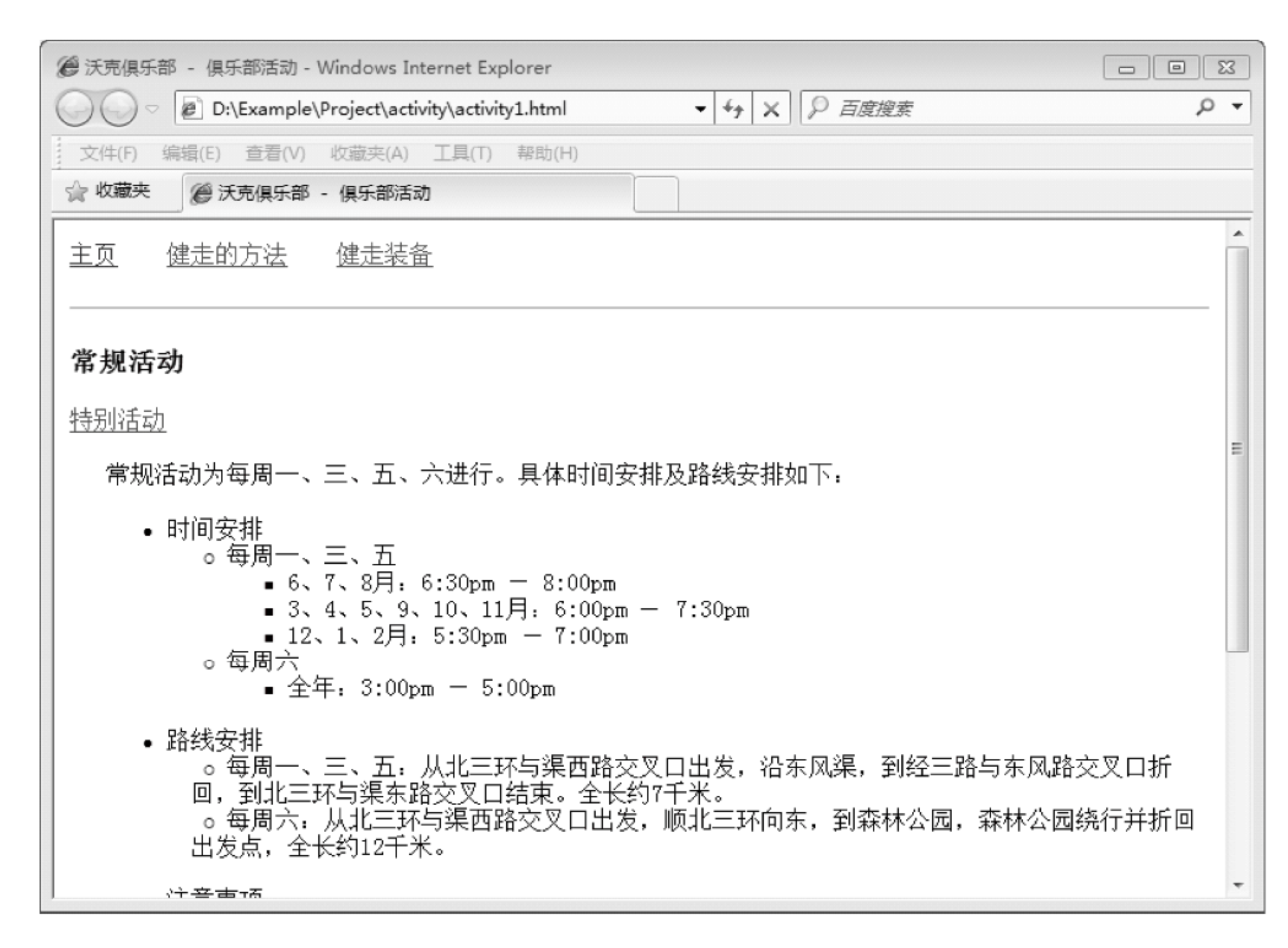


图 2.20 activity1.html 页面效果

提示：这里创建的 4 个网页，主要内容都和 a 元素有关。利用 a 元素在本网页内跳转，或者链接到其他网页，以及自动打开电子邮件客户端并填写好收件人邮箱地址等，尤其重要的是，这 4 个网页分别位于三个不同的目录（文件夹），使用 a 元素链接这些位于不同目录的网页，是读者需要熟悉掌握的技能。

2.4 插入图片

2.4.1 img 标签

例 2.37 插入图片。

```
<html>
<body>
  <img src = "smile.gif" alt = "Big smile face" />
</body>
</html>
```

例 2.37 在浏览器中的效果如图 2.21 所示。

使用<img>标签嵌入一个图像到 HTML 文档中。

img 标签语法：

```
<img src = "file name of image" alt = "description text of image" />
```



图 2.21 img 元素

GIF、JPEG 和 PNG 是网页中最常见的图片格式。

当一个网页被打开，与此同时浏览器会从服务器上获得图片并且将图片显示在网页中。因此，确保图片一直存放与网页有链接的地方。否则，网页的访问者会得到一个无效的链接图标。如果浏览器不能找到图片，会显示无效的链接图标。

要显示一个图片，需要使用 src 属性。src 的属性值是在网页上显示的图片的 URL。因此，可以把 src 看作是一个链接，链接到要使用的图片资源。在例 2.37 中，图片“smile.gif”和文件“例 2.37.html”在相同的目录中。然而，如果图片文件和网页文件在不同的目录中，就需要在 src 属性值中使用之前学过的相对路径，参见项目 2.7。

如果浏览器不能显示图片或者是盲人使用浏览器访问图片，可以使用 alt 属性来描述图片。如果不能看到图片，至少可以看到(听到)alt 中的文字。

提示： src 和 alt 是 img 元素两个必须的属性，其中 src 就像是一个链接，链接到要使用的图片资源上；alt 属性是为了给那些不能看到网页中图像的浏览者提供文字说明，其中包括那些使用本来就不支持图像显示或者图像显示被关闭的浏览器的用户，视觉障碍的用户和使用屏幕阅读器的用户，替换文字是用来替代图像而不是提供额外说明文字的。

2.4.2 使用图片做超链接

img 元素能够做成超链接，使用语法如下：

```
<a href = "url"><img src = "file name of image" /></a>
```

例 2.38 将图片做成超链接。

```
<a href = "../index.html"><! -- click the image to go to homepage -->
```

```
<img src="smile.gif" alt="Go to Homepage" /><!-- show image smile.gif -->
  </a>
```

提示：以图片作为链接实现起来并不复杂，只要把图片元素嵌入到超链接元素内即可。

2.4.3 插入图片综合应用

项目 2.7 在 article1.html 中增加图片。

```
  <html>
    <head>
      <title>沃克俱乐部 - 健走的方法</title>
    </head>
    <body>
    <!--将本网页首部书签设为 top, 本网页其他部分可以用"回首部"来回到首部-->
      <p><a href="../index.html" name="top">主页</a>   
      <!--使用 target 属性,使 article2.html 在新窗口打开-->
        <a href="article2.html" target="_blank">健走装备</a>
      </p>
      <hr />
      <h3>健走的方法</h3>
      <p><a href="#jzyl">健走要领</a></p><!--设定网页内跳转-->
      <div style="text-indent: 1.5em">
        <!--设定图片居右对齐,文字与图片上下左右之间有 15 像素的空白-->
        <img src="../images/walking1.jpg" alt="健走有利身体健康" align="right" hspace
="15" vspace="15" />
        <p>健走即俗称的快走。健走如果步伐适当,其速度也不亚于慢跑。现代的竞走运动中,一
些运动员能达到每公里 3 分 30 秒的速度,一个优秀的竞走运动员,20 公里只需要一个半小时就能完
成。这个速度对于跑友来说,都已经相当快了。与跑步相比,快走通常更持久,而且,如上所述也能达
到很大的强度,足够锻炼心肺功能,同时还能发展腿部、腰部和背部肌肉的力量。</p>
        <p>在我国古代的军队中,就有一种急行军比赛,要求士兵快走完成比赛,而不是跑,当时的
距离是 90 公里,要求在 8 个小时之内。可见,在古代就有用快走锻炼身体的传统。</p>
        <p>健走比慢跑好的地方也有几点,一是不容易对足部、脚踝和膝盖造成伤害,二是没有特
殊的要求,只要不妨碍迈步,普通服装也可以。</p>
        <p>另外,健走和跑步一样,形式也是多种多样的,可以在海边走,也可以在山上走,还可以
在室内走,但健走也有一个标准,那就是时速不能低于 6 公里。</p>
        <p><a name="jzyl">健走的动作要领如下:   </a><a href="#top">
回首部</a></p>
        <ol><!--设定排序列表 ordered list-->
          <li>身体前倾 3~5 度,抬头,肩和背自然下垂,挺胸,收腹。</li>
          <li>两臂必须配合摆动,手臂摆动的高度不得高于胸。</li>
          <li>脚跟着地要柔和,快速滚动到全脚掌着地。</li>
          <li>两脚落地时以内侧为准踩成一条直线。</li>
          <li>不要出现双脚都腾空的现象。</li>
          <li>健走要求走路跨大步、速度敏捷、双臂摆动、抬头挺胸,比慢跑安全,也比散步有效。
</li>
        </ol>
      </div>
      <div style="text-align:center"><!--网页尾部设为文本居中对齐-->
        <hr />
        <p><a href="mailto:webmaster@walkerclub.net">和我联系 Email: webmaster@
```

```
walkerclub.net</a></p>
        </div>
    </body>
  </html>
```

项目 2.7 在浏览器中显示效果如图 2.22 所示。



图 2.22　在网页中增加图片

添加图片以后，网页更加生动与丰富。

提示：img 元素在 HTML 版本中，还有诸如 width、height、hspace、vspace、align 等属性，这些属性用来设定图片大小、图片与周边文字空白间距及对齐等，其中部分属性在项目 2.7 中出现。在 HTML 严格要求中，这些功能可以通过 CSS 方式来完成。

2.5 表格

例 2.39 是一个简单的 HTML 表格，包含两行和两列。

例 2.39　一个简单表格。

```
<table border="1">
  <tr>
      <th>Name</th>
      <th>Date of Birth</th>
```

```
    </tr>
    <tr>
        <td>Jeff</td>
        <td>08/12/1988</td>
    </tr>
</table>
```

表格使用<table>标签定义。一个表格被分为若干行(使用<tr>标签),每个行被分为若干数据单元格(使用<td>标签)。td 的是意思是 table data,是一个数据单元格的内容。一个数据单元格可以包括 text、images、lists、paragraphs、forms、horizontal rules、tables 等。

Table 标签语法:

```
<table border = "pixels">
  <caption>caption text</caption>
  <tr><th>header cell1</th><th>header cell2</th>…</tr>
  <tr><td>cell1</td><td>cell2</td>…</tr>
   ⋮
</table>
```

border 属性是用来显示带有边框的表格。

<caption>标签用于创建表格的标题。

<tr>标签用于在表格中插入一行。所有包含在其中的<td>和<th>标签将会在相同的表格行中显示。

<th>标签用于在表格中插入一个表头单元。表头单元中的内容通常是粗体并且在单元格内居中。

<td>属性用于在表格中创建数据单元格。

现在使用以上标签和属性改进例 2.39,变为一个四行三列带有标题的表格。

例 2.40 带标题的稍复杂表格。

```
<html>
  <body>
    <table border = "1">
      <caption>Students Name List</caption>
      <tr>
        <th>Name</th>
        <th>Date of Birth</th>
        <th>Gender</th>
      </tr>
      <tr>
        <td>Jeff</td>
        <td>08/26/1988</td>
        <td>Male</td>
      </tr>
      <tr>
        <td>David</td>
        <td>11/18/1987</td>
        <td>Male</td>
```

```
        </tr>
        <tr>
          <td>Rose</td>
          <td>12/22/1988</td>
          <td>Female</td>
        </tr>
      </table>
    </body>
  </html>
```

图 2.23 为例 2.40 在浏览器中的显示效果。

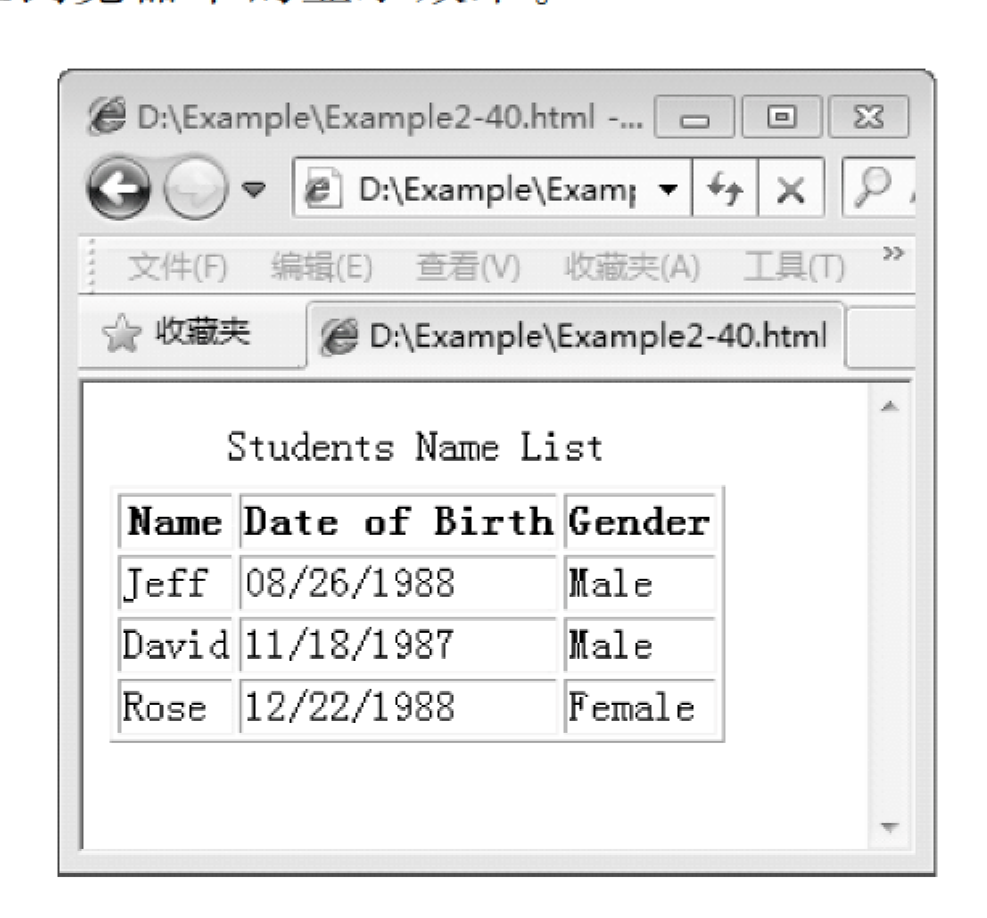


图 2.23 在网页中插入表格

提示：从例 2.40 中可以看到，在网页中做一个四行三列的表格，要写很多代码，如果要做一个有很多数据并且结构复杂的大表格，那将是相当大的代码工作量；另外，<table>标签还有 width，height，cellpacing，cellpadding 等诸多属性来设定表格宽度、高度及数据在表格中的放置方式，这样又增加了表格的 HTML 代码书写的工作量。事实上，表格很少使用纯书写代码的方式来完成，可以利用网页制作工具如 Dreamweaver 来快捷地画出表格，或者用动态网页设计工具 ASP.net 对提取出的数据根据事先设计好的表格自动进行显示。

值得注意的是，由于单元格内可以放置图片、文字、超链等元素，所以表格在网页中常用来建立网页的框架，使整个页面布局分布更规则，并使条目更清晰。

2.6 HTML5

2.6.1 基本 HTML5 文件

为了对 HTML5 有一个基本的认识，首先来看一个最基本的 HTML5 文件。

例 2.41 基本的 HTML5 文件。

```
<!DOCTYPE html>
<html>
<head></head>
<body>
```

```
    <video src = "movie.mp4" width = "320" height = "240" controls>
    Your browser does not support the video tag.
    </video>
</body>
</html>
```

在浏览器上运行结果如图 2.24 所示，播放一个视频文件。



图 2.24 一个最基本的 HTML5 文件

如果读者曾经使用 HTML4(或者 XHTML)实现视频播放工作，对例 2.41 所完成的工作一定有极为深刻的印象，短短的几行代码，就完成了视频的播放。这就是 HTML5 对多媒体文件播放的有力支持，当然，这只是 HTML5 众多强大而新颖的特性之一。

HTML5 对视频和音频的支持将在稍后讲到，这里首先讲一下 HTML5 文件的最基本的架构特征。

仔细看一下例 2.41，这个 HTML5 文件与以往的 HTML4(或 XHTML1)不同之处就是它的第一行。

```
DOCTYPE
```

第一行的!DOCTYPE 是文档类型声明(Document Type Declaration)，它被用来告诉浏览器，应该使用什么样的文档类型定义来解析该文档(或文件)。对 HTML 文件而言，它需要说明 HTML 的版本。!DOCTYPE 要出现在 HTML 文件(无论什么版本)的第一行，在<html>元素之前。

HTML5 问世之前，文档类型声明代码冗长而难记，比如，对 XHTML 1.0 Transitional，其文档类型声明代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

对 HTML5 来说，极大地简化，是如下的代码：

```
<!DOCTYPE html>
```

或者小写格式也是正确的：

```
<!doctype html>
```

HTML5 的写法简单而直观。

2.6.2 定义页面结构布局

HTML5 新增加的标签中，引人注目的部分，包括引入常用的页面布局元素。现在，来详细了解在页面布局中，HTML5 都添加了哪些元素。

1. header 元素

用户访问网站，第一眼看到的，就是位于最上面的 header 元素。header 元素通常包括介绍性的信息以及网站及网页导航信息。在实际设计中，虽然 header 元素经常放在页面的顶部，但并不意味着 header 元素不能放在页面的其他位置。根据文章的需要、帖子的需要，header 元素也可以放在左面、右面，甚至是内容的下面。

2. section 元素

section 元素是按主题分组的内容，通常带有标题。比如，作为教科书，本小节就是一个 section，专门用于讲解页面结构布局这个主题的内容，其 section 标题是“2.6.2 定义页面结构布局”。

section 元素常用于以下的页面设计：

- 选项卡界面中的每一个的选项卡。
- “服务条款”页面中的每一个部分。
- 在线新闻网站的新闻分类，如文章可以分类为体育新闻 section，国际新闻 section，及经济新闻 section 等。

3. article 元素

顾名思义，一个 article 即是一篇描述完整的独立内容。对于 article 元素的使用场合，有如下建议：

- 论坛帖子。
- 杂志报纸文章。
- 一条博客。
- 用户提交的评论。

4. nav 元素

nav 元素即是其英文词汇 navigation（导航）所表达的含义：一组导航链接。可以确定地说，每个网站项目中都有 nav 元素出现。

nav 元素可以在页面中出现多次。比如，主导航栏和页面内导航都可以放到 nav 元素内。

5. aside 元素

本元素是与页面内容分离而又附属于页面的部分。aside 元素可用于以下设计：

- 特定独立的内容（仍然附属于文章或节）。
- 整个网站的侧边栏内容。

实际应用中，aside元素可用在侧边栏、广告区等设计中。需要说明的是aside元素并不是只能放在“边”栏位置，它也可以放在其他位置。

6. footer 元素

类似于Word中的页脚，footer元素则是Web页面中的页脚。当然，设计实践中，footer元素在一个页面中可以出现多次，作为文章的脚，作为节的脚，作为页面的脚。

Footer元素通常包含版权信息、相关链接、作者(开发者)信息等适合放在整体内容末尾部分的信息。与header、aside元素类似，虽然footer常常放在整体部分脚部，但设计者也可以放在其他认为合适的位置。

7. 构造页面布局

HTML5新增的页面布局元素介绍完毕。现在利用这些元素构造一个页面。

Web页面首部有一个header元素，包括Logo、标题和宣传词(如品牌标语)，该header元素中还包含nav元素，整个网站的链接在该元素中定义。

首部下面，网站主要内容分为两列，其中左列是边栏，右列是文章内容。

页面脚部是footer元素，该布局元素包含版权、作者、网站备案信息等。

参见例2.42，使用HTML5布局元素完成了一个基本的网页布局设计。

例2.42 使用HTML5布局元素设计网页。

```
<!DOCTYPE html>
<html>
<head></head>
<body>
<header>
<nav></nav>
</header>

<div id = "maincontent">
<div id = "sidebar">
    <aside></aside>
    ...
</div>
<div id = "article">
    <article></article>
    ...
</div>
</div>

<footer>
<section id = "copyright">
    <section></section>
</section>
<section id = "author">
    <section></section>
```

```
</section>
</footer>

</body>
</html>
```

例 2.42 完成的布局图如图 2.25 所示。

在例 2.42 中，类似 id = “maincontent”这样的代码将在第 3 章 CSS 中讲到；同时，由于本例中选择符 id 都没有进行具体的 CSS 定义，因此，图 2.25 只是一个概念图，具体实现，需要 CSS 的配合才可以完成。

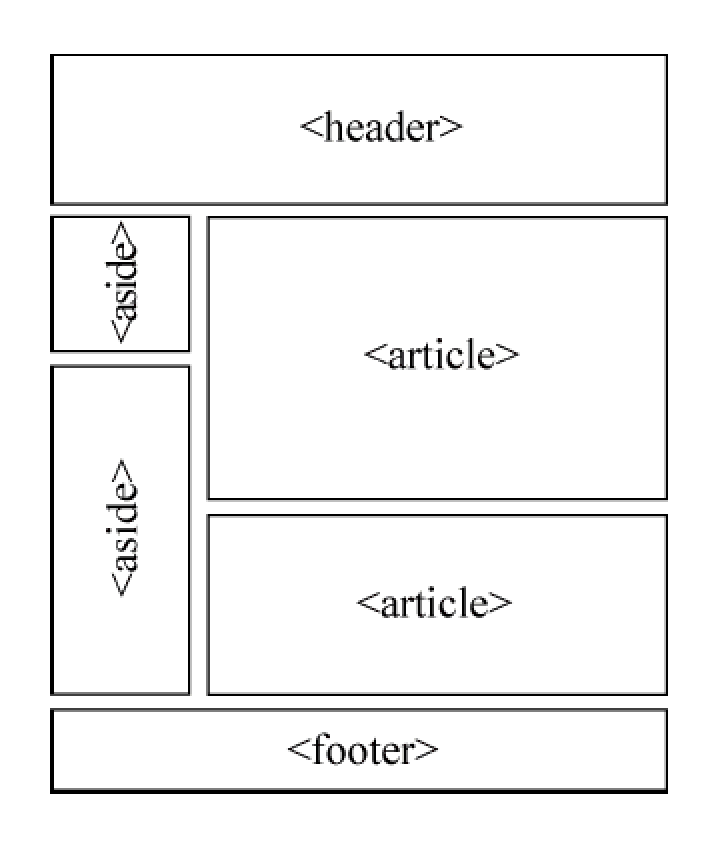


图 2.25　使用 HTML5 布局元素设计网页

说明：HTML5 中页面架构布局的元素，其核心思想是用于替代传统 HTML 布局中使用的 DIV 元素。在学习第 3 章 CSS 时，<div id = “header”></div>代码，可以由<header></header>替代，以保证更简洁更规范的页面布局写法。

当然，对于已经习惯于使用 DIV 和 CSS 自主进行页面布局设计的人来说，转而使用 HTML5 需要一个过程。比如，<div>与<section>有什么区别？什么时候使用<div>，什么时候使用<section>？在这里给出的建议是：如果不是非常确定使用某一个特定的 HTML5 布局元素，仍使用<div>的方式完成页面布局设计。

2.6.3　视频和音频播放

HTML5 出现之前，Web 页面没有播放视频和音频的标准，那时的视频及音频是通过插件(如 Flash)进行播放，而且，不同的浏览器支持的插件也各不相同。

Adobe 公司的插件 Flash 播放器由 Adobe 独家控制，不对 Web 开发社区开放。HTML5 引入视频和音频播放元素，使多媒体播放成为 Web 页面无缝结合的一部分。使用 HTML5，用户不必下载第三方的多媒体播放软件，就可以直接播放多媒体内容。

1. 视频播放

参见前述的例 2.41，即 HTML5 的第一个例子，其中 video 元素展示了播放视频文件的功能。现在详细学习 video 元素的使用方法。

参见例 2.41，使用 video 元素时，建议把 width 和 height 两个属性明确赋值。当为视频

设置宽度和调试值时，网页加载时，就会为视频保留这个空间。如果没有事先赋值，浏览器不知道视频的宽和高，就无法预留空间，载入视频时，再把视频需要的空间挤出来，这样会改变已经加载的网页布局，影响用户体验。

video 元素中的 controls 属性加入播放控件，如播放、暂停、音量调节等。

video 元素内的文本内容是为不支持 video 元素的浏览器准备的，如果浏览器不支持 video 元素，这些文本内容就会显示出来。

video 元素语法和 img 元素语法很类似，需要指定源文件的位置(src)及文件所占空间大小(width，height)等属性。例 2.41 中的核心代码如下(有适当删减)：

```
<video src = "movie.mp4" width = "320" height = "240"></video>
```

其中 src 指定视频源文件位置，width 和 height 指定视频播放的宽与高。

video 语法如下：

```
<video src = "url" width = "pixels" height = "pixels" attributes> text </video>
```

属性说明：

src：必需。source 英文缩写，指定视频源文件位置。建议使用相对路径。

width：可选。视频宽度，以像素值表示。

height：可选。视频高度，以像素值表示。

虽然可以设置视频的宽和高，但视频本身的比例不会受影响。比如：在上述例子，例 2.41 中，如果视频实际是 320×200(宽×高)，标签中设置的是 320×240，那么视频将保持高度为 200 像素不变，在 240 像素的空间中垂直居中。显而易见，这样做是为了保持视频原本比例，不被 video 属性设置的宽和高扭曲变形。

controls：可选。该值为布尔类型，直接添加属性名称，不设定值。有这个属性时，视频界面显示播放功能按钮，如播放、暂停、停止、快速定位、音量调节等，如图 2.24 所示。实际代码如下(选自例 2.41)：

```
<video src = "movie.mp4" width = "320" height = "240" controls>
</video>
```

autoplay：可选。布尔类型。有此属性时，含有页面视频的页面加载后，视频自动播放。实际代码如下：

```
<video src = "movie.mp4" width = "320" height = "240" controls autoplay>
</video>
```

loop：可选。布尔类型。有此属性时，视频将循环播放。实际代码如下：

```
<video src = "movie.mp4" width = "320" height = "240" controls autoplay loop>
</video>
```

对多种视频格式的支持。

如果所有浏览器能够统一支持一种类型的视频格式，当然是很好的解决办法。事实上，问题没有这么简单。目前，主流浏览器支持 3 种视频格式：MP4、WebM 和 Ogg。表 2.2 列出了主流浏览器对这 3 种视频格式的支持情况。

表 2.2 支持 HTML5 视频格式的浏览器版本

Browser	MP4	WebM	Ogg
IE	9+	—	—
Chrome	3+	6+	3+
Firefox	3.5+	4+	3.5+

为了在 video 元素中包含多种格式的视频文件，在 video 元素中嵌套了另外一个元素，名为 source 元素，该元素用于指定视频源文件，source 元素的功能就相当于 video 元素中的 src 属性，当使用 source 指定多个格式的视频源文件时，video 元素中的 src 属性就不再使用。

考虑到当前的主流浏览器对视频格式的支持，多个格式视频文件的支持代码如下。

例 2.43 同时指定多个格式的视频文件。

```
<!DOCTYPE html>
<html>
<body>
    <video width="320" height="240" controls>
        <source src="movie.mp4" type="video/mp4">
        <source src="movie.webm" type="video/webm">
        <source src="movie.ogg" type="video/ogg">
        Your browser does not support the video tag.
    </video>
</body>
</html>
```

例 2.43 中，movie.mp4、movie.webm、movie.ogg 这三个文件具有相同的视频内容，只是格式不同，这样可以保证该网页在主流浏览器中都能够被正常播放。这也说明，HTML5 标准化还有很长的路要走。

同样，在例 2.43 中，每个 source 元素中都必须包含 type 的属性，这个属性指定 MIME 类型，用于告诉浏览器用什么样的 MIME 类型解析源文件。3 种视频源文件对应的 MIME 类型如表 2.3 所示。

表 2.3 视频格式的 MIME 类型

视频格式	MIME 类型
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

2. 音频播放

除了视频播放的可视部分外，上小节讨论的关于视频播放的相关内容大都适合于音频播放。因此，音频播放内容可以类比视频播放内容学习与掌握。

类似于视频播放 video 元素，其中的 src、controls、autoplay、loop 属性都适用于音频播放 audio 元素。audio 元素在使用 controls 属性时，播放器操作界面才能够显示出来（与

video 类似)。当然,播放器操作界面也可以通过 JS 脚本的方式访问,需要时显示,不需要时隐藏。audio 元素嵌套 source 元素,支持多种音频格式。浏览器支持的音频格式如表 2.4 所示,音频格式对应的 MIME 类型如表 2.5 所示。

表 2.4 支持 HTML5 音频播放的浏览器版本

浏 览 器	MP3	Wav	Ogg
IE	9+	9+	—
Chrome	3+	6+	3+
Firefox	—	3.5+	3.5+

表 2.5 音频格式对应的 MIME 类型

Video Format	MIME-type
MP3	audio/mpeg
Wav	audio/wav
Ogg	audio/ogg

在 audio 元素中,嵌套的普通文本,是为不支持 audio 元素的浏览器显示使用。如例 2.44,使用 IE9(支持 audio 元素)浏览时,可以正常播放音频,使用 IE8 及以下版本浏览时,无法播放音频,可以看到"Your browser does not support the audio element."文本显示在浏览器中。

例 2.44 音频播放。

```
<!DOCTYPE html>
<html>
<body>
    <audio controls>
        <source src="music.wav" type="audio/wav">
        <source src="music.mp3" type="audio/mpeg">
        Your browser does not support the audio element.
    </audio>
</body>
</html>
```

说明: HTML5 在无缝化多媒体播放方面迈出了决定性的一步。网站开发设计人员可以不需要第三方插件完成多媒体的播放工作。在实际学习应用中也要看到,由于不同浏览器对多媒体格式支持的不同,不是一行代码可以应用于所有的浏览器上,因此,在设计中,需要考虑到所有的主流浏览器,要写多行代码,同样,要把同一个多媒体文件转换为多种格式,才可以保证多媒体文件在主要的浏览器都能够正常播放。

小结

HTML5 是在 HTML4 基础上,为了更好的可读性,直接支持多媒体效果,并可以运行在更多的设备之上。HTML4 的标签仍然可以使用,在组织网页架构时,注意利用 HTML5 标签。

HTML是网页设计的基础，虽然现在有许多网页设计工具可以自动生成HTML代码，但对HTML深入的掌握，可以直接使用代码方式更细节化地设计调整网页。

习题

1. 下列哪一行代码是符合HTML规范的？

   A. <p>A short<i> paragraph.</i></p>

   B. <p>A<i> short</i> paragraph. </p>

   C. <p>A short<i> paragraph. </p>

   D. <p>A short<i> paragraph.</i>

2. 以下哪个程序不能用于编辑HTML文档？

   A. 计算器　　B. 记事本　　C. 写字板　　D. Word

3. HTML所有元素都需要关闭么？

4. 在HTML中，<. br />和
哪一个是正确的换行标签？

5. 如何在新窗口打开链接？

   A. <a href=”#” new>

   B. <a href=”#” target=”_blank”>

   C.

   D.

6. 怎样可以由页面底部快速跳转到页面顶部？

7. 在HTML5中引入了专门用于播放视频文件的video元素。如果在一个网页设计中，使用video元素的src属性成功调用源文件myvideo.ogg，并使用Chrome浏览器测试播放成功，那么，这个网页上的视频文件能够在所有主流浏览器上都播放成功么？

第3章 CSS

3.1 CSS 简介

HTML 不包含格式化文本的标签。HTML 的目的是要定义一个文本的内容，比如：

```
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```

当<font>标签和颜色属性被添加到 HTML 3.2 格式，它成为了 Web 开发人员的噩梦。把字体和颜色信息添加到每个网页中，使设计大型网站变成一个漫长而昂贵的过程。

为了解决这一问题，万维网联盟(World Wide Web Consortium，W3C)创建了层叠样式表(Cascading Style Sheets，CSS)。在 HTML 1.0 中，可以从 HTML 文档删除所有格式，储存在一个单独的 CSS 文件中。现在所有的浏览器都支持 CSS。

内容与表现的分离是 HTML 的基本设计原则之一。把展示与内容分离可以把相同的内容发给不同的客户端浏览器，让这些客户端决定如何用最合适的方式表现出来。毕竟，一个手机浏览器没有与火狐等桌面浏览器相同的功能。

因此，HTML 文件应着重于文档内容是什么，而不是显示出什么样子。HTML 元素如何显示由 CSS 来进行定义。目前，HTML + CSS(或称为 DIV + CSS)已经成为 Web 的一个标准的设计方法。

CSS 是网页设计界的一场革命。CSS 的具体好处包括：

- 用一个单一的样式表控制多个文档的布局。
- 更精确的布局控制。
- 对不同媒体类型(屏幕、打印等)采用不同布局。
- 众多先进和尖端的技术。

提示： HTML 最初只是为了显示网页的内容，不是用来美化网页的布局。但在 20 世纪末的浏览器大战中，浏览器开发商为了使自己的浏览器显示出更美观的网页，纷纷在 HTML 中加入可以美化网页显示的属性，这使得网页设计人员不得不设计对应某一特定浏览器的网页，导致一个网站会有针对不同浏览器的不同版本。为解决这一问题，W3C 推出了 CSS，并做出明确分工：HTML 负责内容的构建，CSS 负责网页的布局、美化与展示。现在所有的主流浏览器都支持 CSS，也形成了当前网页设计的标准化方法：HTML + CSS (DIV + CSS)。

3.2 将 CSS 插入网页

插入一个样式表有三种方式：

- 行内样式。
- 内部样式表。
- 外部样式表。

3.2.1 行内样式

行内样式的语法：

```
<tag style = "property: value; property: value; …">
```

tag：HTML 标签的名称。

style：样式属性，style 参数可以应用于除了 basefont，param 和 script 的任意<body>内的元素。样式属性可以包含任何 CSS 属性。

冒号“：”是用来分隔属性和值，分号“；”是用来分割属性的定义。

例 3.1 行内样式的使用。

```
<html>
  <head>
    <title>Inline Style</title>
  </head>
  <body>
    <p style="font:30pt; color:red">This inline style specifies the paragraph's font size as 30pt and font color as red.</p>
    <p>This paragraph has no inline style.</p>
  </body>
</html>
```

例 3.1 在浏览器上的显示效果如图 3.1 所示。

由于行内样式将内容与展示混合在一起，它失去了引入样式表的优势，尽量少用这种办法。

提示：行内样式可以单独控制一个标签的样式。因为行内样式把网页的格式化与需要显示的内容(即 HTML 标签)混在一起，实际上违背了最初设计 CSS 的初衷，也失去了 CSS 特有的优势，所以这种方法应尽量少用。

第 2 章的 Example 2.25 在<div>标签中使用了这种行内样式的方法，当时看来还属于比较先进的方法，当学完本章后，就会发现有更简捷方法来完成同样的工作。

3.2.2 内部样式表

单一的文件要具有独特的风格应使用内部样式表，内部样式通过使用<style>标签，在 HTML 页面的<head>部分中定义，如下所示。

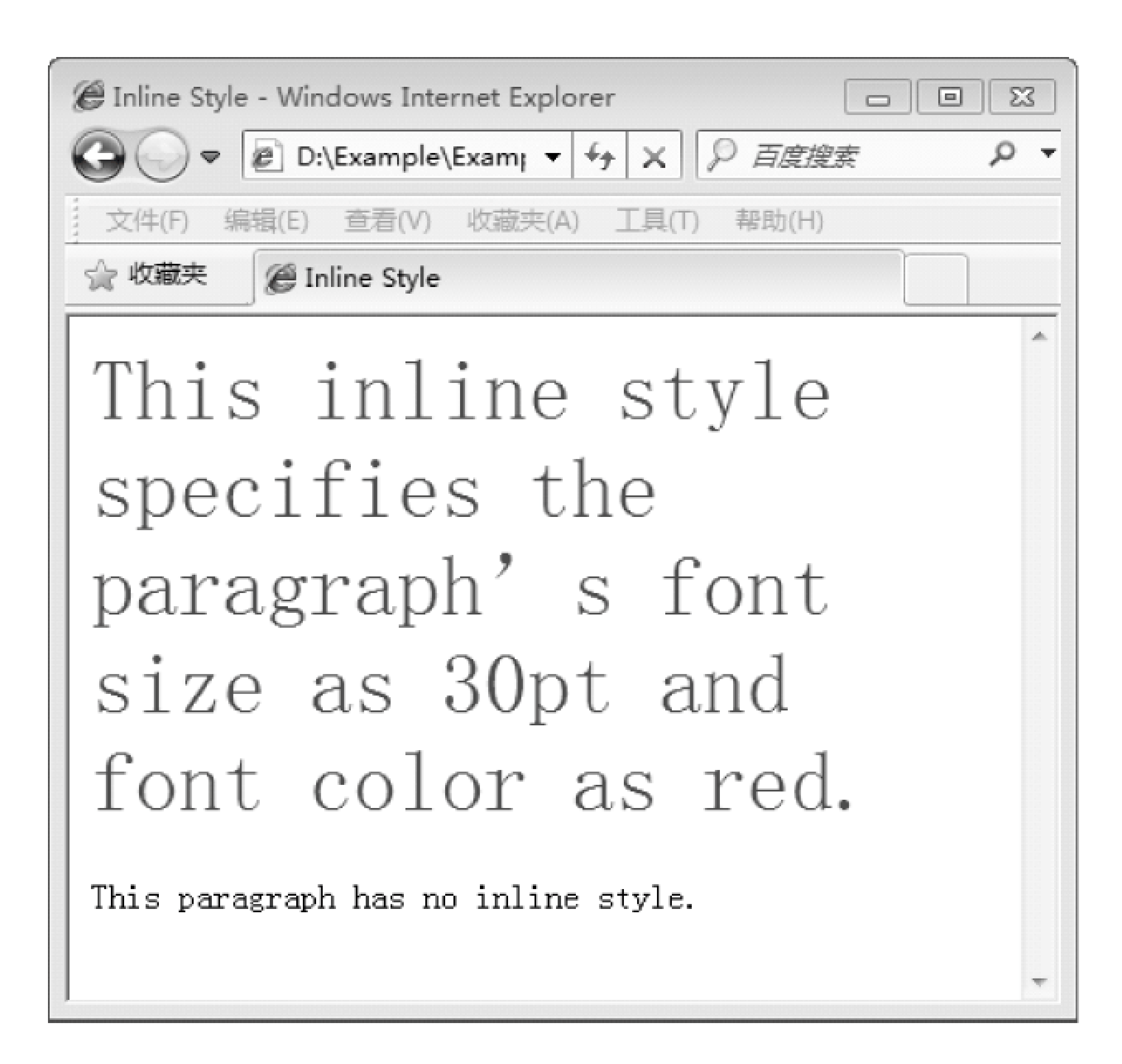


图 3.1 行内样式

内部样式的语法：

```
<style type = "text/css">
<! --
  selector1{ property: value; property: value; … }
  selector2{ property: value; property: value; … }
  …
-->
</style>
```

style：在此标签内定义样式。

type：根据 CSS 语法定义指定样式(type = "text/css")。

<!--…-->：HTML 的注释标记，用来使旧版的浏览器可以使用，因为它们可能不知道 CSS。

selector：通常是 HTML 元素/标签来定义(将在本章稍后部分学习)。

属性和值之间用冒号，并用大括号包围。

例 3.2 使用内部样式表。

```
<html>
 <head>
    <style type = "text/css">
    <! --
    h1.mylayout {border - width: 1; border: solid; text - align: center;color:red}
       -->
  </style>
</head>
<body>
  <h1 class = "mylayout"> This heading uses the style.</h1>
  <h1 > This heading doesn't use the style.</h1>
```

```
</body>
</html>
```

例 3.2 在浏览器上的显示效果如图 3.2 所示。

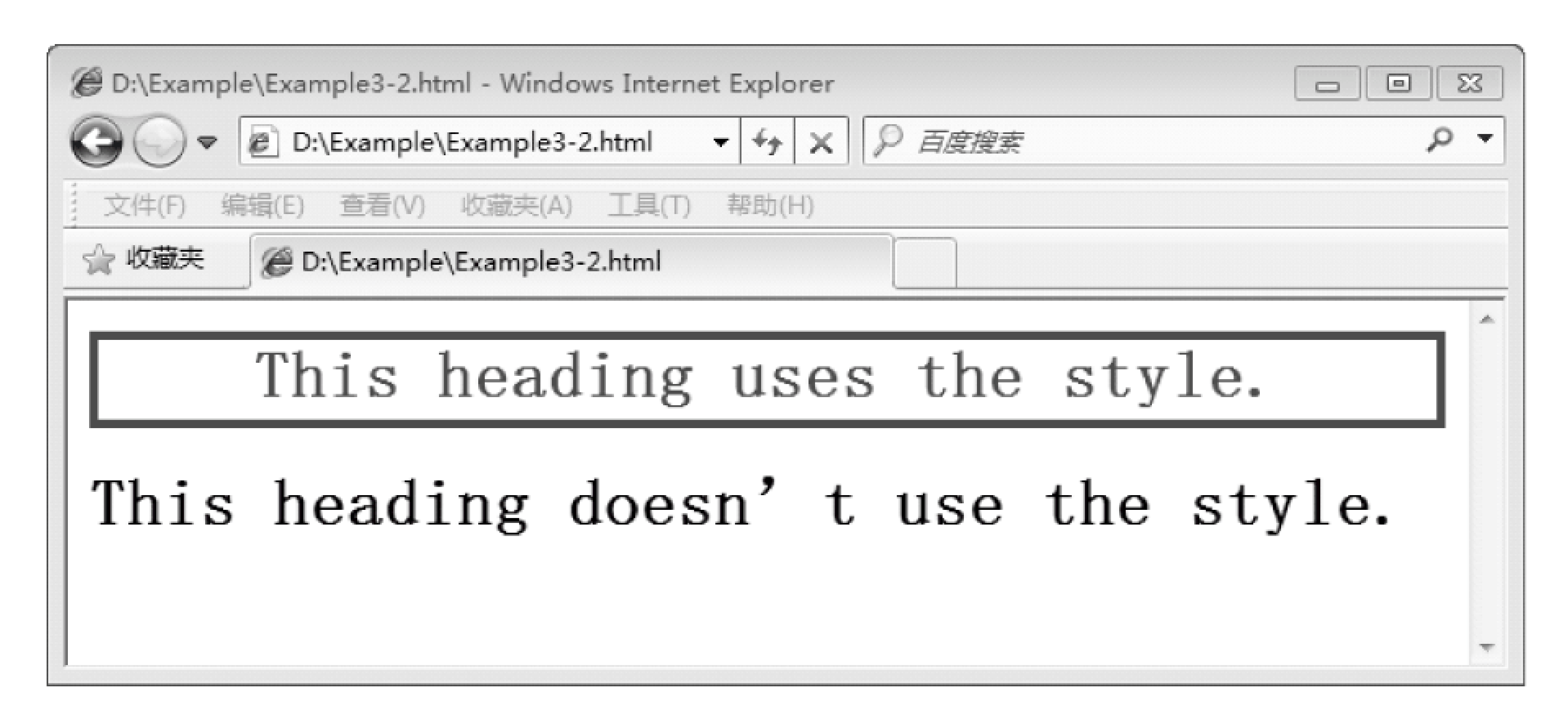


图 3.2 内部样式表

内部样式表的优点：

- 内部样式只影响它们所在的页面。如果在一个大型网站工作，需要测试样式，在将网页加载到整个网站之前，内部样式表是一个很有用的工具。内部样式表允许测试当前文档的样式，而不会破坏整体网站的其他页面。
- 内部样式表可以使用类和 ID。和行内样式不同，内部样式表仍然可以利用类、ID 和其他元素关系。
- 内部样式表不要求上传多个文件。当使用那些只有一个 HTML 文档去编辑内容，比如电子邮件或小型应用系统时，它是非常有用的。将文档的样式与文档本身放在一起，就知道是什么样式在影响文件。
- 内部样式具有比外部样式表更高的优先级。这取决于外部样式的加载顺序。Web 页面设计人员可以将内部样式放置在文件头部的合适位置。如果内部样式被放置在连接到外部的样式表的后面，内部样式在层叠上有更高的优先级。

内部样式表的缺点：

- 只能影响它们所在的网页。如果几个文档要使用相同的样式，需要在每个页面上重复放入这些内部样式(或连接到一个外部样式表上)。
- 内部样式表的加载时间长。浏览器每次加载页面时，页面内的内部样式都需要加载和解析。而外部样式表由浏览器缓存，这样，当第一次加载外部样式表之后，页面加载不再需要加载样式表，极大节省了时间。

提示：关于选择符(selector)及其使用将在 3.3 节进行详细的讲解。这里需要更多关注的是内部样式所在位置(head 元素内)和 HTML 的注释标签(<!-- ….. -->)的使用。注释标签是为了避免旧版本的浏览器不支持 CSS，所以特意把 style 元素内容以注释形式表示，这样对于不支持 CSS 的浏览器，会自动忽略此段内容。

本小节讨论了内部样式的优点和缺点，由于读者尚未进行外部样式的学习，可能对此理解并不深刻。学完外部样式及层叠(cascading)概念后，本小节关于内部样式优缺点的讨论

会完全理解。

3.2.3 外部样式表

当样式应用于许多页面时，外部样式表是不错的选择。使用外部的样式表，整个 Web 站点的外观可以通过改变一个.css 文件完成。

外部样式表创建语法与内部样式表相似。也许更简单一些，只需要选择符和声明。外部样式表语法是：

```
selector1{ property: value; property: value; … }
selector2{ property: value; property: value; … }
…
```

将这些代码保存到一个扩展名为.css 文本文件中。

例如，使用记事本创建一个文件名为 layout.css，键入例 3.3 的内容到这个文件中。

例 3.3 创建一个外部样式表文件 layout.css。

```
h1.mylayout {border-width: 1; border: solid ; text-align: center; color:red}
```

外部样式表文件被创建好以后，将其链接到网页上，有两种方法来实现这种链接。

1. 链入

使用 HTML <link>标签链接外部样式表。<link>标签放置在<head>部分。

链入标签的语法：

```
<link rel="stylesheet" type="text/css" href="styles.css" />
```

rel：指定当前文档和链接文件之间的关系，这里是一个样式表链接。

type：指定链接的文档的 MIME 类型。

href：指定链接的文档的位置，这里是到外部样式表(.css)文件的路径。

例 3.4 链入外部样式表文件。

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="layout.css" />
  </head>
  <body>
    <h1 class = "mylayout"> This heading uses the style.</h1>
    <h1> This heading doesn't use the style.</h1>
  </body>
</html>
```

例 3.4 在浏览器显示结果如图 3.2 所示，与例 3.2 相同。

不过，作为一个内部样式表，例 3.2 只能影响它所在的页面；作为一个外部的样式表文件，即例 3.3 的 layout.css，可应用于网站内的任何 HTML 文档。

2. 导入

在内部样式表中使用导入的外部样式表，这样外部样式表导入后，内部样式表的属性不

受任何影响。

导入外部样式表的语法：

```
<style type = "text/css">
<!--
  @import url("styles1.css");
  @import url("styles2.css");
  internal style sheet declaration
-->
</style>
```

styles1.css，styles2.css 等都是导入到本网页的外部样式表文件。

注意：

(1) @import 声明必须置于<style>元素的开始部分，然后才是内部样式表声明。

(2) @import 声明的顺序决定样式表如何层叠。

(3) @import 声明后的分号不能省略。

可以导入许多的外部样式表来维护网站。

例 3.5 导入外部样式表文件。

```
<html>
  <head>
    <style type = "text/css">
    <!--
    @import url("layout.css");
    -->
    </style>
  </head>
  <body>
    <h1 class = "mylayout"> This heading uses the style.</h1>
    <h1>This heading doesn't use the style.</h1>
  </body>
</html>
```

例 3.5 在浏览器中显示如图 3.2 所示，和例 3.2、例 3.4 相同。

提示：外部样式表是把样式定义单独创建一个外部文件中，扩展名为.css。然后在 HTML 文档中使用<link>或@import 的方式将外部样式表文件引入文档，对网页进行布局展示。这是 CSS 推荐的方式，也是 CSS 的优势所在，使用一个外部样式表文件，可以完成对整个网站的网页外观、布局、图像、字号、字体等的设计。

如本小节所述，引入外部样式表有两种方法，两种方法的使用效果是一样的，读者可以选取其中一种方法使用即可。

3.2.4 层叠

层叠样式表或 CSS 的设置能够使许多属性影响到同一个元素。其中一些属性与另一些属性可能会发生冲突。比如，段落标签上的字体颜色可能设置的是红色，之后又有一个设置将这个段落标签的字体颜色设为蓝色。浏览器如何知道这个段落最终是什么颜色？这是

由层叠决定的。

1. 三种样式的样式表

(1) 作者样式。这是由网页设计者创建的样式表,也是多数人所想到的 CSS 样式表。

(2) 用户样式。用户样式表由网页的用户创建,这允许用户能更好地控制页面的显示。

(3) 浏览器样式。Web 浏览器将样式应用到页面帮助显示页面。

上述每种样式有不同的权重。默认情况下,作者样式是第一位,用户的样式次之,最后是浏览器样式。唯一的例外是在用户样式中的!important 规则,它比作者样式更优先(有更高的权重)。

2. 层叠顺序

为解决冲突,Web 浏览器使用下列排序,以确定哪些样式将优先使用。

(1) 首先,寻找元素的所有声明和指定的媒体类型(正确显示在各类媒体,如在屏幕上、在纸上或移动电话上)。

(2) 再看看它是来自什么样式表。如上所述,作者样式第一位,然后是用户,然后浏览器。有!important 的用户样式比有!important 的作者样式有更高的优先级。

(3) 一个选择符越是具体,它会得到越高的优先级。例如,一个“div. co p”的样式比一个“p”标签有更高的优先级。

(4) 最后,按它们被定义的顺序排列。在文档中后面被定义的规则比前面被定义的有更高的优先级。导入的外部样式表规则被认为出现在内部样式表中的规则之前。

3. !important 规则

层叠意味着样式按照浏览器读取它们的顺序来执行。第一个样式首先被应用,然后是第二个。因此,如果一个样式出现在样式表的顶部,然后在样式表下部被改变,那么被实际应用的是该样式的第二个实例,而不是第一个。例如,在下面的样式表的例子中,该段文本将显示为黑色,即使第一个样式属性是红色的。

例 3.6 最近的样式优先级最高。

```
<html>
  <head>
    <style type = "text/css">
    <! --
      p { color: #ff0000; }
      p { color: #000000; }
     -->
    </style>
  </head>
  <body>
    <p>This is a paragraph.</p>
  </body>
</html>
```

!important 规则是有特权的层叠，它保证最重要的规则永远被应用。一个有!important 规则的属性，无论这一规则它在 CSS 文档中何处出现，都将被应用。因此，如果设计者要确保一个属性始终应用，可以添加!important 属性到这个标签。因此，为了使该段文字总是显示为红色，将上面的例子改变如下。

例 3.7 使用!important 规则。

```
<html>
  <head>
    <style type = "text/css">
    <!--
      p { color: #ff0000 !important }
      p { color: #000000; }
    -->
    </style>
  </head>
  <body>
    <p>This is a paragraph.</p>
  </body>
</html>
```

4. 用户样式中的!important

!important 的规则也用来帮助网页浏览用户应付那些让他们使用起来很困难的网页。通常，如果一个用户也定义一个样式表来浏览网页，用户样式表将服从于网页作者的样式表。但是如果用户标记一个样式为!important，那个样式会替代网页作者样式表，即使网页作者用!important 标记他们的规则。

这是 CSS2 上的一个变化。在 CSS1 中，!important 作者样式表的规则优先于!important 用户样式表。CSS2 改变了这点，用户样式表具有优先权。

说明：层叠优先级按次序从高到低依次为：

(1) 标记为!important 的用户样式；

(2) 标记为!important 的作者样式；

(3) 作者样式；

(4) 用户样式；

(5) 浏览器默认样式。

对于设计人员来说，作者样式有三种：行内样式(Inline Style)、内部样式(Internal Style)和外部样式(External Style)，根据规定，这三种样式的优先级别从高到低依次为：

(1) 行内样式；

(2) 内部样式；

(3) 外部样式。

样式表的层叠也可以从继承性来理解，样式表的继承规则是：外部的元素会保留下来，由这个元素所包含的其他元素继承；所有在元素中嵌套的元素都会继承外层元素指定的属性值；有时会把多层嵌套的样式叠加在一起，除非另外更改；多个样式对一个元素的定义发生冲突时，以最后定义的为准(没有使用!important 规则情况下，见本小节的两个例子)。

层叠是CSS设计重要的概念，网站设计人员必须对此概念有明晰的理解。作为设计人员，在使用作者样式时，可以遵守最佳实践（见下一小节），杜绝行内样式的出现，尽量少用内部样式，在只使用外部样式的情况下，可以使得样式定义冲突减少，但并不能完全消失。因为外部样式表中定义的多个元素样式，在HTML文档中由于不同的嵌套关系，不可避免地会产生定义冲突。何况还有用户样式及浏览器样式这些网站设计者所不能掌控的样式，也会和作者样式发生冲突。因此在设计时，需要对CSS的层叠有一个前瞻性的考虑。

3.2.5 CSS最佳实践

CSS已经成为网页样式与布局设计事实标准，既然大多数人都使用它，就需要关心如何把它用好。有三种方法使用样式表，虽然它们都有不同的目的，但是使用它们的形式有好有坏。了解CSS的网页设计的最佳实践将确保网页完美无缺。

1. CSS最佳实践

最佳实践是已被证明是最有效的，对工作有最大回报的设计和制作网页的方法。CSS最佳实践可以在以下方面帮助改善网站：

（1）将内容与设计分开。CSS的主要目标之一是消除HTML中的设计元素并把它们放在其他地方以便设计者维护。这意味着，设计师只需要维护网站的外观展示，而不必做一个内容开发人员。

（2）维护方便。网页设计中最容易被遗忘的元素之一就是维护。网站总是变化的——从网站上的外观、内容及内部链接都在变化。让CSS文件在一个核心位置使得它更容易维护。

（3）保证网站更易访问。使用CSS样式可以使网站更容易被残疾人使用，更容易被搜索引擎找到。

（4）网站将更久地紧跟最新技术。使用CSS最佳实践，当网站设计环境变化时，这些设计仍然能够工作，具有相当的灵活性。

2. 什么才是CSS最佳实践

使用外部样式表而不是使用行内样式。外部样式表拥有CSS最佳实践的所有优点，并且易于使用。如果必须在一个特定的HTML文档内放入样式，把它们放在内部样式表中，并把内部样式表的放在这些文件的<head>部分。这样，至少它们仍与内容分开。避免对网页的任何内容使用行内样式。

提示：了解CSS设计的最佳实践，设计网页则会事半功倍。综上所述，读者也更明白为什么在网页设计中要避免出现行内样式（Inline Style），因为它违反以上所有的最佳设计方法。既然这样，问题出来了：为什么CSS还要留着行内样式？根据前一小节的层叠概念，行内样式是设计者所能掌控的样式中优先级最高的样式，因此它所产生的效果不会被其他样式所覆盖。根据这一特性，网页设计人员可以在测试样式效果时使用它，起到立竿见影的效果。但要注意的是，效果测试成功后，需要把这部分内容做到外部样式表文件中，在正式的网页文件中不要出现行内样式。

同样，对一个规范严谨的网站，内部样式（Internal Style）也要尽量避免，因为它也只是

部分地做到了网页内容与网页设计相分离。学习 CSS 的目标就是：建立一个统一的外部样式表，在网页中调用它。

本章在进行 CSS 举例的时候，会较多地用到内部样式(Internal Style)。这是由于为了说明举例效果，样式定义都比较简短，所以就放在<head>部分的内部样式中进行定义，在一个文件中可以较方便地对比 CSS 的样式定义代码与其在 HTML 内容上产生的效果，与本章所提倡的实际网页设计中尽可能使用外部样式并不矛盾。同时，由于本教材有大量例子，为使读者注意力集中在所讲解的内容，同时也为节省篇幅，正常网页文件中应该有的 DTD 声明及<head>部分的<title>标签，在示例中一般不写，但在沃克俱乐部的项目举例中会完整表现出来。

3.3 类选择符和 id 选择符

3.3.1 CSS 语法

CSS 的语法由三部分组成：选择符、属性和值。

```
selector {property: value}
```

选择符通常是 HTML 元素/标签。属性和值之间用冒号，并用大括号包围，如：

```
body {color: black}
```

如果有多个属性被指定，每个属性必须用分号隔开。下面的例子显示了如何定义一个段落中心对齐，文本颜色为绿色：

```
p {text - align: center; color: green}
```

如果该值是多个单词，用引号括起来：

```
p {font - family: "sans serif"}
```

为了使样式定义更具可读性，可以在每一行声明一个属性，如：

```
p
{
text - align: center;
color: red;
font - family: Arial
}
```

多个选择符可以一次定义，每个独立的选择符以逗号分开。在下面的例子中所有的标题元素都被一次性定义，所有标题内容以蓝色显示文字颜色。

例 3.8 一次性定义多个选择符。

```
<html>
  <head>
    <style type = "text/css">
    <! --
```

```
        h1,h2,h3,h4,h5,h6
        {
        color:blue
        }
      -->
      </style>
    </head>
    <body>
      <h1>This is heading 1</h1>
      <h2>This is heading 2</h2>
      <h3>This is heading 3</h3>
      <h4>This is heading 4</h4>
      <h5>This is heading 5</h5>
      <h6>This is heading 6</h6>
    </body>
</html>
```

CSS 注释

注释是用来解释 HTML 代码的，在以后编辑源代码时帮助网页的设计者。注释将被浏览器忽略。一个 CSS 注释以“/*”开头，以“*/”结尾，如：

```
/* This is a comment */
p
{
text-align:center;
/* This is another comment */
color:red;
font-family: Arial
}
```

提示：在浏览器中看一下例 3.8 效果，读者可以初步了解 CSS 设计的优点与理念。在 HTML 中，每一个标签都对应一定的内容，比如<p>标签，是设定一个段落，<h#>是设定一个 1 号到 6 号的标题，虽然也有对齐、字体及颜色等功能，但需要在每个标签内进行定义，应用起来非常烦琐。在 CSS 中，则统一对这些标签进行对齐、字体、字号、色彩等布局展示方面进行定义，在随后的<body>中使用的相应的 HTML 标签，就使用这种定义，一次定义，反复使用，多处使用，这就是 CSS 的优势所在。

从现在开始，除非举例的特别需要，<body>中的 HTML 标签，不再使用诸如对齐、颜色等属性设置，这些工作都交由 CSS 来完成，HTML 标签专注于内容属性的设置，如 href、src 等。

3.3.2 类选择符

其实，类选择符的概念已经显示在例 3.2 中。

用类选择符，不同的样式可以被定义到同一个 HTML 元素上。

比如想在 HTML 文件中有两种类型的<h1>：一个右对齐的<h1>，一个居中的<h1>。下面是它如何使用样式做到这一点。

例 3.9 对 h1 元素使用类选择符。

```
<html>
  <head>
    <style type = "text/css">
    <!--
      h1.right {text-align:right}
      h1.center {text-align:center}
    -->
    </style>
  </head>
  <body>
    <h1 class="right">This heading is on the right</h1>
    <h1 class="center">This heading is in the middle</h1>
  </body>
</html>
```

例 3.9 在浏览器上的显示效果如图 3.3 所示。

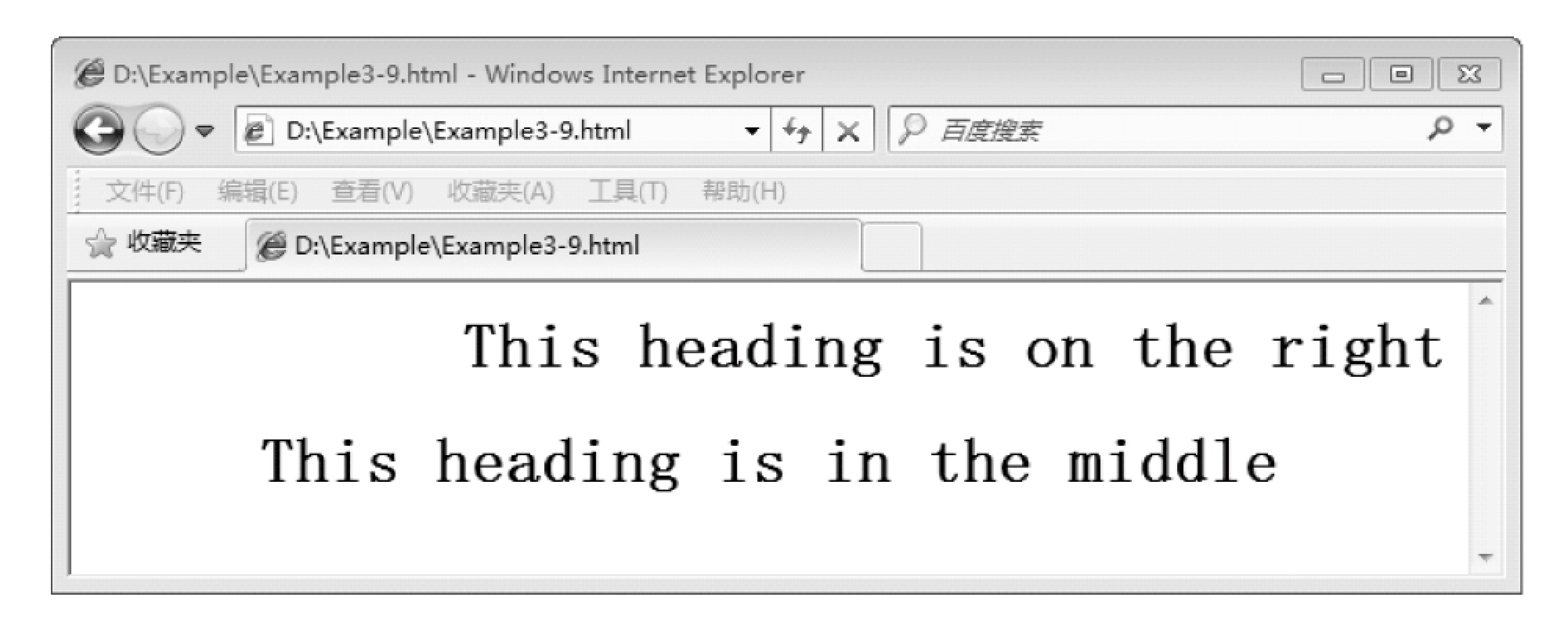


图 3.3 使用类选择符定义相同的元素

类选择符的语法如下：

```
tag1.classname1 {property: value; property: value; … }
tag2.classname2 { property: value; property: value; … }
…
tagN.classnameN { property: value; property: value; … }
```

tag.classname 仍然被称为选择符。类名定义类选择符的名称；名称可以是任何字母的组合或以字母开始的字母和数字的组合。不要用数字作为类名称的开头。

在 HTML 文件中使用类选择符的语法：

```
<tag class = "classname1 classname2 …"> content </tag>
```

要在一个给定的元素上应用多个类,类之间要用空格隔开。

例 3.10 一个元素调用多个类。

```
<html>
  <head>
    <style type = "text/css">
```

```
    <! --
      h1.right {text - align:right}
      h1.border {border - width: 1; border: solid}
    -- >
    </style>
  </head>
  <body>
    <h1 class = "right border">This heading is on the right</h1>
  </body>
</html>
```

例 3.10 在浏览器上的显示效果如图 3.4 所示。



图 3.4　在一个元素上应用多个类

如果想让一个类被所有的 HTML 元素使用，可以将类名前边标签省略。在下面的例子中，所有引用 class="center"的 HTML 元素都将居中对齐。

例 3.11　多个标签调用同一个类。

```
<html>
  <head>
    <style type = "text/css">
    <! --
      .center {text - align:center}
    -- >
    </style>
  </head>
  <body>
    <h1 class = "center">This heading 1 is in the middle</h1>
    <h4 class = "center">This heading 4 is in the middle too</h4>
    <p class = "center">This paragraph is also center - aligned.</p>
  </body>
</html>
```

例 3.11 在浏览器上的显示效果如图 3.5 所示。

说明：类选择符定义方式有两种：有标签的类选择符，其适用范围只限于该标签所包

含的内容;省略标签的类选择符(注意在定义时类名前边不要忘了加点“.”),可以使任意HTML标签套用这种预先定义好的类样式。读者可以尝试将例3.11的居中对齐样式定义改为:

```
h4 .center {text - align:center}
```

其他内容不变,用浏览器显示,发现只有h4居中,其他两个元素居左(缺省对齐方式)。这种省略HTML标签的类选择符是最常用的定义方法,影响范围大,使用方便灵活。

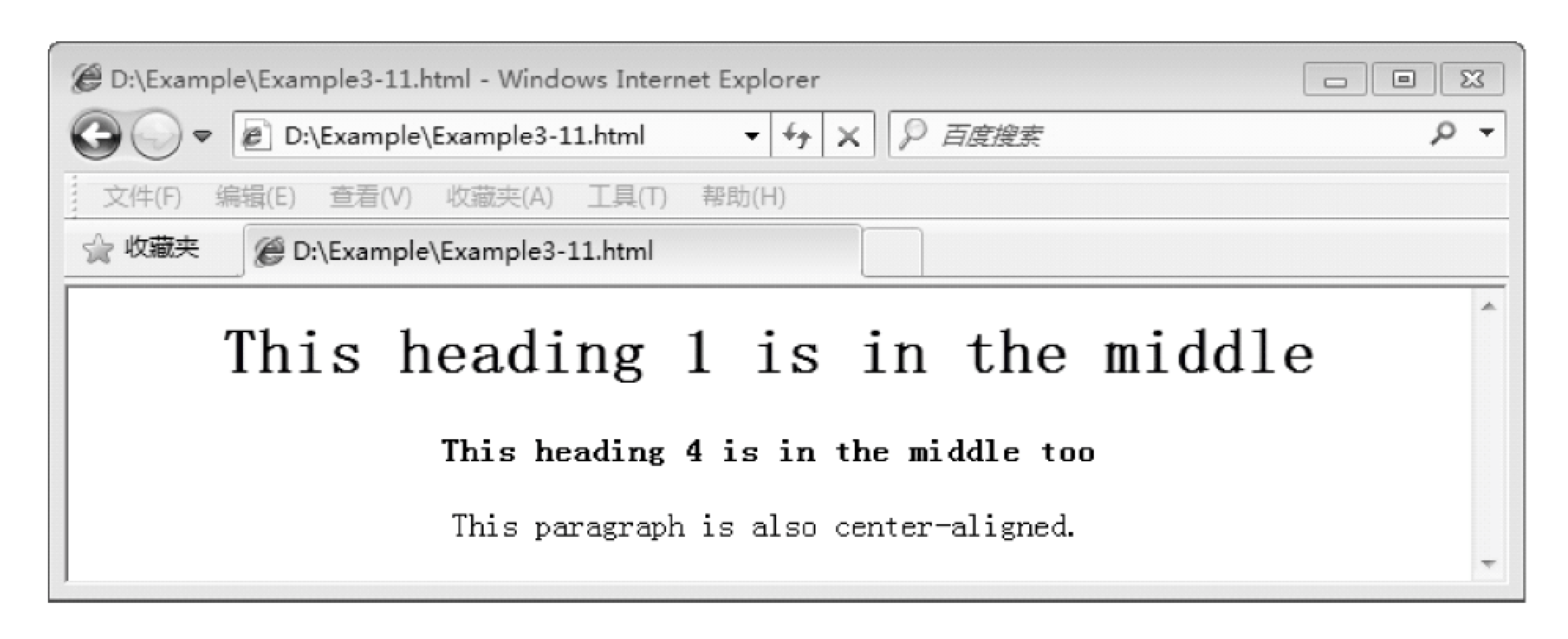


图3.5 类选择符被多个元素调用

3.3.3 id 选择符

和类选择符一样,id选择符有两种类型的定义:含有HTML标签和不含HTML标签。含有HTML标签的id选择符的语法:

```
tag1 # idname1 {property: value; property: value; … }
tag2 # idname2 { property: value; property: value; … }
…
tagN # idnameN { property: value; property: value; … }
```

不含HTML标签的id选择符的语法:

```
# idname1 {property: value; property: value; … }
# idname2 { property: value; property: value; … }
…
# idnameN { property: value; property: value; … }
```

id的名称可以是任何字母的组合或以字母开始的字母和数字的组合,不要用数字作为id名称开头,因为在一些浏览器中它无法正常工作。

在HTML文件中调用id选择符的语法:

```
< tag id = "idname"> content </tag >
```

id 选择符与类选择符对比

除句号(.)和磅符号(#)区别之外,id选择符和类选择符之间似乎没有差异。在讨论区别之前,先来知道CSS为什么要选择这些名字。

id:一个人的身份(ID)是独一无二的。

类：在一类中有很多人。

所以，

id 是独一无二的：

- 每个元素只能有一个 id。
- 每个页面只能有一个元素使用该 id。

类不是独一无二的：

- 同一个类可用于多个元素。
- 几个类可用于同一元素。

标准规定，任何给定的 id 名只能被页面或文件引用一次。从以往的经验看，CSS 布局中 id 最经常被正确应用于页面布局。因为通常每页只有一个菜单、一个通栏标题、一个内容窗格。

每页只使用一次时用 id，每页使用一次或多次时使用类。

例 3.12　id 被 HTML 文件多次调用。

```
<html>
  <head>
    <style type = "text/css">
    <!--
      #center {text-align:center}
    -->
    </style>
  </head>
  <body>
    <h1 id="center">This heading 1 is in the middle</h1>
    <h4 id="center">This heading 4 is in the middle too</h4>
    <p id="center">This paragraph is also center-aligned.</p>
  </body>
</html>
```

例 3.12 在浏览器上的显示为图 3.5 所示，和例 3.11 的显示相同。但是，如果用 W3C 的标准验证例 3.12（网址：http://validator.w3.org/），错误信息的显示如图 3.6 所示。

提示：从语法上来看，id 选择符的声明及使用和类选择符是类似的，只是名称和使用的符号有所变化。但两者有实质上的区别。id 选择符，顾名思义，是独一无二的，所以它在一个网页中只能出现一次，一个元素也只能引用一个 id 选择符，类选择符则可以多次使用。

虽然在例 3.16 中，一个网页多次使用 id 选择符并且显示结果与例 3.15 使用类选择符时的显示结果相同，但它不符合 W3C 标准，所以标准性测试就出现了如图 3.6 的错误信息，同时也明确地给出了错误原因。

通过以上分析，可以看出由于类选择符可以使用一次，也可以使用多次，灵活性要好于 id 选择符，所以一些网页设计者偏好使用类选择符。如某个样式定义肯定在页面中只出现一次，比如页头、页脚等，使用 id 选择符是一个不错的办法。

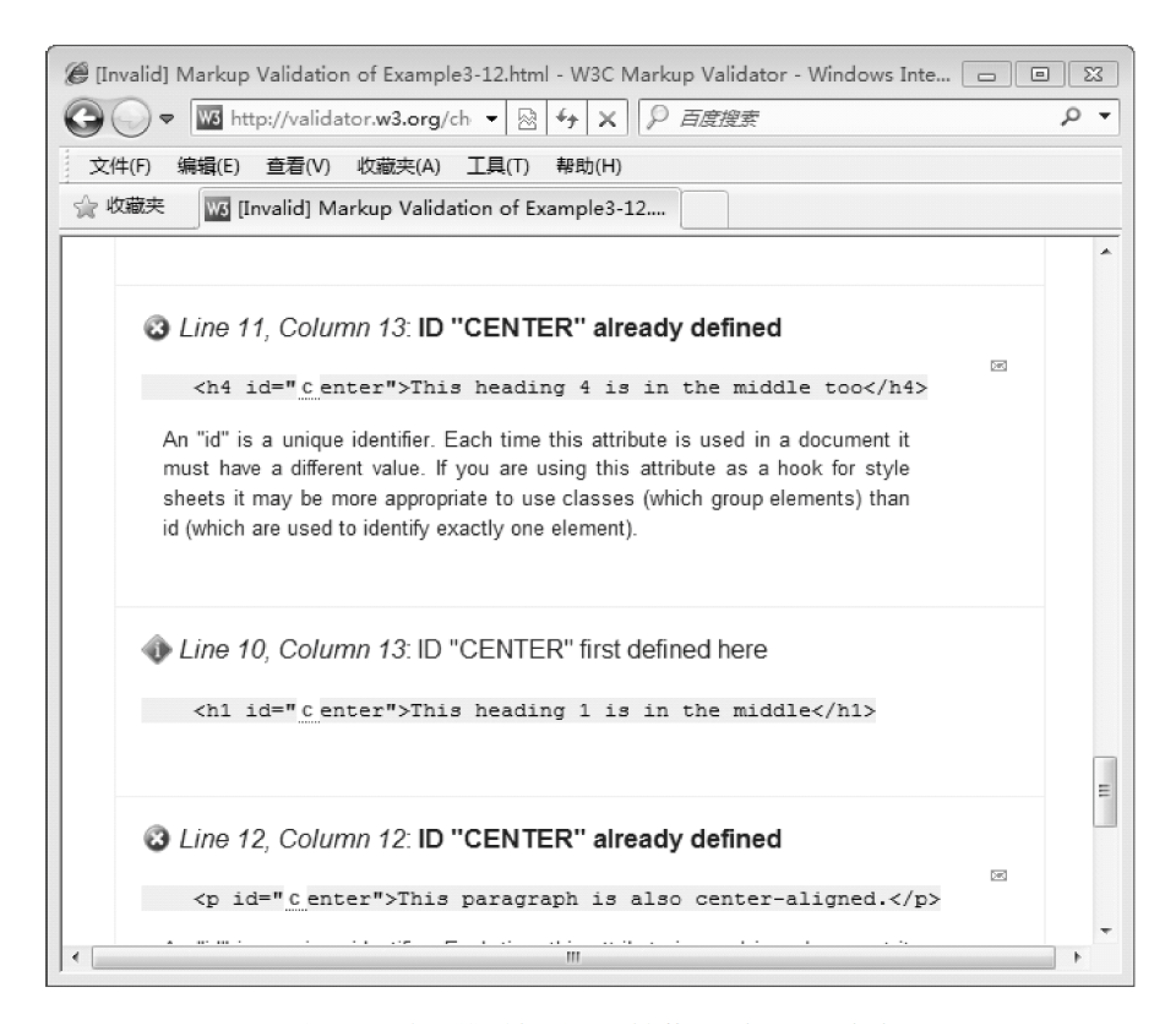


图 3.6 验证错误提示：id 被使用(定义)了多次

3.4 CSS 常用属性

3.4.1 CSS 字体

CSS 字体属性定义了字体名称、字体加粗、大小及文字样式等。

1. 字体名称

属性：font-family

值：Times、Georgia、Arial、宋体、隶书、楷体…

一个文本的字体由字体名称属性来设置。字体名称属性可以容纳几个字体名称作为后备字体。如果浏览器不支持第一个字体，它会尝试下一个字体。

注意：如果一种字体的名字是多个单词，它必须用引号，如字体名称 Times。

字体名称列表中指定多个字体名称时，使用逗号分隔：

```
p{font - family:"Times",Georgia,Serif}
```

2. 字体大小

属性：font-size

值：length | % | small | medium | large | smaller | larger | …

字体大小属性设置文字的大小。

能够管理的文字大小在 Web 设计中很重要。但是，字体大小调整不应用来使段落看起来像标题，或者使标题看起来像段落。始终使用正确的 HTML 标签，用<h1>到<h6>来设置标题，<p>来设置段落。

字体大小的值可以是一个绝对或相对的值。

绝对大小（mm、cm、in、pt）：

- 设置文本到指定大小。
- 不允许用户在所有浏览器中改变文字大小（不适宜用户浏览，适宜于打印输出）。
- 当输出的物理尺寸是已知的时候，绝对大小非常有用。

相对大小（em、ex、px）：

- 根据周围元素设置大小。
- 允许用户改变浏览器文字的大小。

下面的例子用像素单位设置字体大小。

例 3.13 设置文字大小。

```
<html>
  <head>
    <style type = "text/css">
    <! --
       h1 {font - size:40px}
       h2 {font - size:30px}
       p {font - size:14px}
     -->
    </style>
  </head>
  <body>
    <h1>This is heading 1</h1>
    <h2>This is heading 2</h2>
    <p>This is a paragraph.</p>
  </body>
</html>
```

例 3.13 在浏览器上的显示效果如图 3.7 所示。

Internet Explorer 无法在例 3.13 中调整字体大小。为了避免 IE 中文字大小调整的问题，许多设计者使用 em 而不是像素。一单位 em 的大小是由 W3C 推荐的。1em 等于当前字体大小。在浏览器中文字的默认大小为 16px。所以，1em 默认大小为 16px。从像素到 em 的大小使用这个公式计算：像素/16 = em。

例 3.14 使用 em 单位设置字体大小。例 3.14 在浏览器显示如图 3.7 所示。用 em 设置的文字大小与例 3.13 的像素相同，但 Internet Explorer 可以调整字体大小。

例 3.14 使用 em 单位设置文字大小。

```
<html>
  <head>
```



图 3.7 通过使用像素或 em 单位的字体大小

```
    <style type = "text/css">
    <! --
      h1 {font - size:2.5em} /* 40px/16 = 2.5em */
      h2 {font - size:1.875em} /* 30px/16 = 1.875em */
      p {font - size:0.875em} /* 14px/16 = 0.875em */
    -->
    </style>
  </head>
  <body>
    <h1>This is heading 1</h1>
    <h2>This is heading 2</h2>
    <p>This is a paragraph.</p>
  </body>
</html>
```

3. 字体风格

属性：font-style

值：normal (default) | italic | oblique

字体风格属性主要是用于指定斜体文本。

这个属性有三个值：

- normal——该文本正常显示(默认值)。
- italic——该文本显示为斜体。
- oblique——该文本是“斜体”(oblique 类似斜体,但支持它的浏览器较少)。

例 3.15 字体风格。

```
<html>
  <head>
    <style type = "text/css">
    <! --
      .normal {font - style:normal}
      .italic {font - style:italic}
```

```
        .oblique {font - style:oblique}
      -->
    </style>
  </head>
  <body>
    <h1 class = "normal">This is heading 1, normal</h1>
    <h2 class = "italic">This is heading 2, italic</h2>
    <h3 class = "oblique">This is heading 3, oblique</h3>
  </body>
</html>
```

例 3.15 在浏览器上的显示效果如图 3.8 所示。

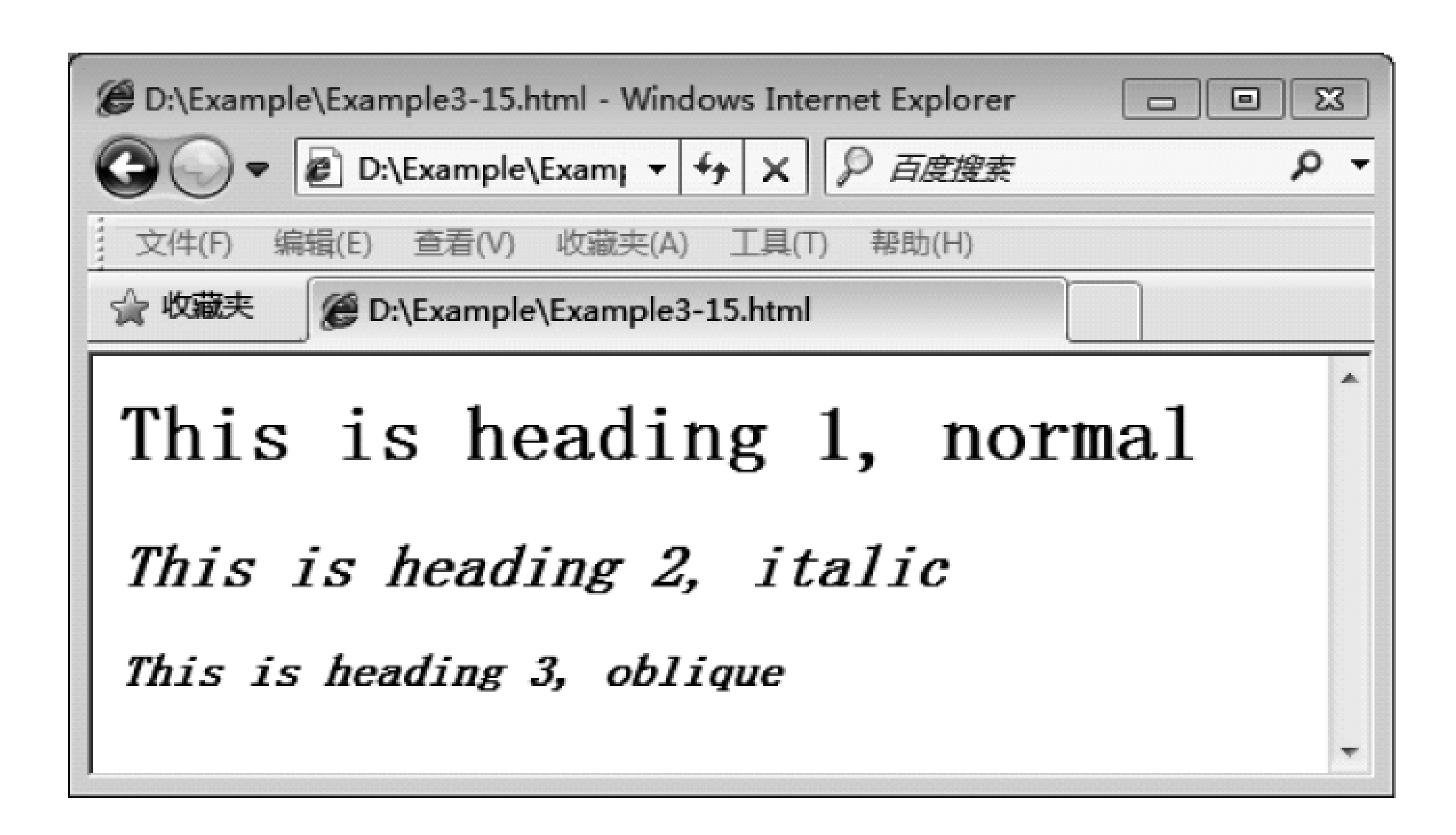


图 3.8 CSS 字体风格

5. 字体粗细

属性：font-weight

值：normal（default）| bold | bolder | lighter | 100 | 200 | 300 | …

字体粗细属性指定了字体的粗细。它主要的值有两个：normal（default）| bold，使用如下：

```
.bold{font - weight:bold}
.normal{font - weight:normal}
```

6. 字体变化

属性：font-variant

值：normal（default）| small-caps | inherit

字体变化属性决定一个文本是否应显示小型大写字母。它主要的值有两个：normal（default）| small-caps，使用如下：

```
.small
{
font - variant:small - caps;
}
```

7. 字体简写属性

属性：font

值(按顺序)：font-style font-variant font-weight font-size/line-height font-family

字体简写属性将所有的字体属性集于一体。可设置的属性是(按顺序)：字体风格、字体变化、字体粗细、字体大小或行高、字体名称。行高属性设置行与行间的空间。

如果上述值在简写中有缺少，例如“font:100% verdana;”，则以默认值代替缺少的值。字体简写属性可以这样使用：

```
.s1 {font: italic normal bold 1.2em Arial}
```

提示：字体的属性在本小节已经全部列出，但属性值却有很多，比如字体名称，读者可以根据需要选择相应的字体。关于字体属性有几点需要说明：

(1) 字体大小(font-size)有许多单位，有绝对大小(如 mm、cm、pt 等)，有相对大小(如 em、px 等，px 是相对设备大小)，这种属性单位概念在最后一小节讲解。网页设计中，W3C 推荐使用 em 单位。

(2) 字体变化(font-variant)是将英文设定成全部小型大写字母，此属性对中文没有作用。

(3) Font 属性是把前边所学的所有字体属性一次性完成声明，在实际网页设计中，推荐使用这种办法设定字体属性。要注意必须按文中指定的属性顺序声明。

3.4.2 CSS 文本

CSS 文本属性定义文本的外观。

1. 文本对齐

属性：text-align

值：left | center | right | justify

text-align 属性用于指定水平对齐的文本，文本可以居中(center)、或向左(left)或向右(right)对齐，或者是两端对齐(justify)。如果 text-align 被设置为 justify，那么每行将被拉伸以使每一行具有相同的宽度，而且左右边距也是整齐的(如杂志和报纸)，text-align 属性可以这样用：

```
.date {text-align:right}
p {text-align:justify}
```

2. 文本修饰

属性：text-decoration

值：none | underline | overline | line-through | blink

text-decoration 属性用于修饰文本，它经常用于为凸显设计意图而去掉超链接默认的下划线。

例 3.16 去除超链接默认的下划线。

```
<html>
  <head>
    <style type="text/css">
    <!--
      a {text-decoration:none}
    -->
    </style>
  </head>
  <body>
    <a href="mailto:webmaster@walkerclub.com">Click to Email me.</a>
  </body>
</html>
```

图 3.9 是例 3.16 在浏览器中的显示效果，Click to Email me. 的下划线被去掉。

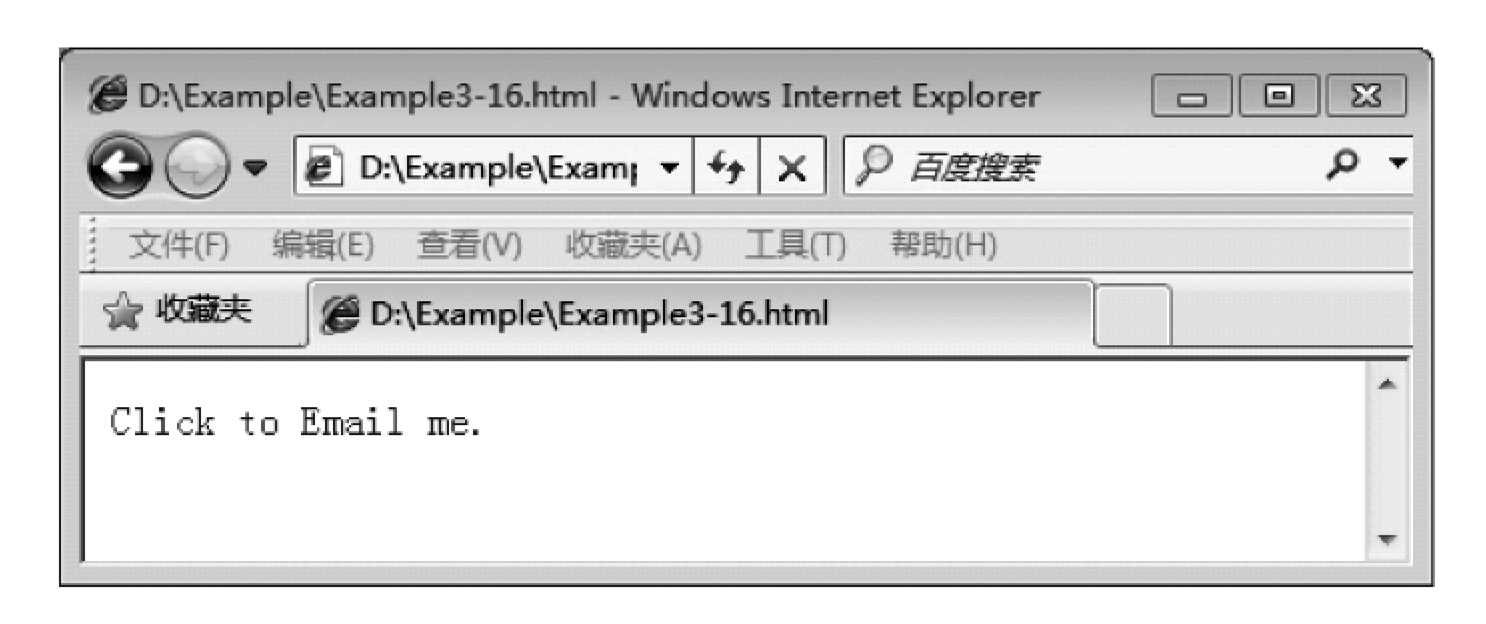


图 3.9 用 text-decoration 属性去掉了超链接的下划线

3. 文本缩进

属性：text-indent

值：长度| %

text-indent 属性用于缩进第一行文本。与 font-size 属性类似，推荐使用相对长度单位 em，text-indent 属性使用方法：

```
div{text-indent: 1.75em}
```

4. 文本颜色

属性：color

值：颜色名称| RGB |十六进制数

color 属性用于设定文本颜色。颜色可以被定义为：

名称-颜色名称，如 red

RGB - RGB 值，如 rgb(255,0,0)

十六进制数-十六进制数，如＃ff0000

color 属性使用方法：

```
h1 {color: #ff0000}
```

```
p {color: green}
```

5. 字符间距

属性：letter-spacing

值：normal（默认）|长度

letter-spacing 属性用来设置字符间的间距，当设置非默认的字符间距时推荐使用相对长度单位 em。

下面对沃克俱乐部网页使用 CSS 思想进行设计。

项目 3.1 CSS 文件 layout. css。

```
div.content{text-indent:1.5em}              /* 设定首行缩进 */
.center{text-align:center}                  /* 设定文本居中对齐 */
hr.topwidth{width:80%}                      /* 设定顶部横线宽度 */
```

项目 3.2 主页文件 index. html。

```
<html>
  <head>
    <title>沃克俱乐部欢迎您</title>
    <!--链接外部样式表文件 layout.css-->
    <link rel="stylesheet" type="text/css" href="layout.css" />

  </head>
  <body>
    <div class="center"><!--调用文本居中对齐类-->
      <h1>咱们走着瞧 - 欢迎光临沃克俱乐部</h1>
      <hr class="topwidth" /><!--调用横线宽度类-->
      <p>
        <a href="introduction/article1.html" target="_blank">健走的方法</a> 
  
         <a href="introduction/article2.html" target="_blank">健走装备</a> 
  
        <a href="activity/activity1.html" target="_blank">俱乐部活动</a>
      </p>
    </div>
    <h3>走路,简单有效的健身之道</h3>
    <div class="content"> <!--调用首行缩进类-->
      <p>走路是人类本能的身体活动,人类身体结构就是为步行设计的。问题是,随着汽车等代步
工具的普及,人们"好逸恶劳"的条件和理由越来越充足,以致基本的走路功能都有些退化了。根据美
国疾病控制与预防中心的研究,有高达 75%的美国人每天运动量不到 30 分钟,大部分人几乎成天坐
在椅子上,是不折不扣的"沙发土豆"。反观我们中国的现实情形,随着经济的发展,汽车的大量使用
及其它因素,导致缺乏锻炼、体形肥胖的人越来越多。</p>
      <p>走路健身又健脑。我国自古就有"走为百练之祖"的健身经验谈,其中传统医学认为,双脚
是人体的健康之根。走路刺激脚底穴位,能舒筋通络,活血顺气,强身健体。同时,现代运动医学研究
也证实:走路时,骨骼、肌肉、韧带、神经末梢都要参加运动,从而促进血液循环,调节大脑皮层的活动
功能,促使身体各种激素分泌,使人心情愉悦。走路防病并延缓衰老。最新的医学研究表明,一周健
步走 7 小时以上,可以降低 20%乳腺癌、30%心脏病和 50%糖尿病的罹患率,而中老年人每天散步
2.4 公里以上,心脏病发作率将降低 50%。</p>
    </div>
```

```
    <div class = "center"><! -- 再次调用文本居中对齐类 -->
      <hr />
      <p><a href = "mailto: webmaster@walkerclub.net">和我联系 Email: webmaster@
walkerclub.net</a>
      </p>
    </div>
  </body>
</html>
```

提示： 文本属性很多，作为教科书，这里只列出了常用的一些属性。全部的文本属性请参阅 CSS 参考手册。

注意本小节的项目 3.1(样式表文件 layout.css)和项目 3.2(HTML 文件 index.html)。在这个网页设计中，已经摒弃了在 HTML 标签通过添加属性对文本进行对齐、缩进设置的办法(参见项目 2.3)，而使用了 CSS 的设计理念，最后达到的效果和第 2 章项目 2.3 相同。这也是进行 CSS 学习时所要掌握的设计思想。

3.4.3 CSS 背景

CSS 背景属性是用来为元素定义背景效果。

1. 背景颜色

属性：background-color

值：颜色名称| RGB |十六进制

background-color 属性指定元素的背景颜色。定义整个页面的背景颜色需要在 body 选择符中定义。

在下面的例子中，h1、p 和 div 元素都有各自的背景颜色。

例 3.17 设置背景色。

```
<html>
  <head>
    <style type = "text/css">
    <! --
      h1{background - color:orange}
      p{background - color:rgb(224,255,255)}
      div{background - color: # b0c4de}
     -->
    </style>
  </head>
  <body>
    <h1>Background - color of heading 1</h1>
    <div>
    <br /><br /><br />
    This is a text inside a div element.
    <p>This paragraph has its own background color.</p>
    This text is still in the div element.
    <br /><br /><br />
    </div>
```

```
  </body>
</html>
```

例 3.17 在浏览器中的显示效果如图 3.10 所示。

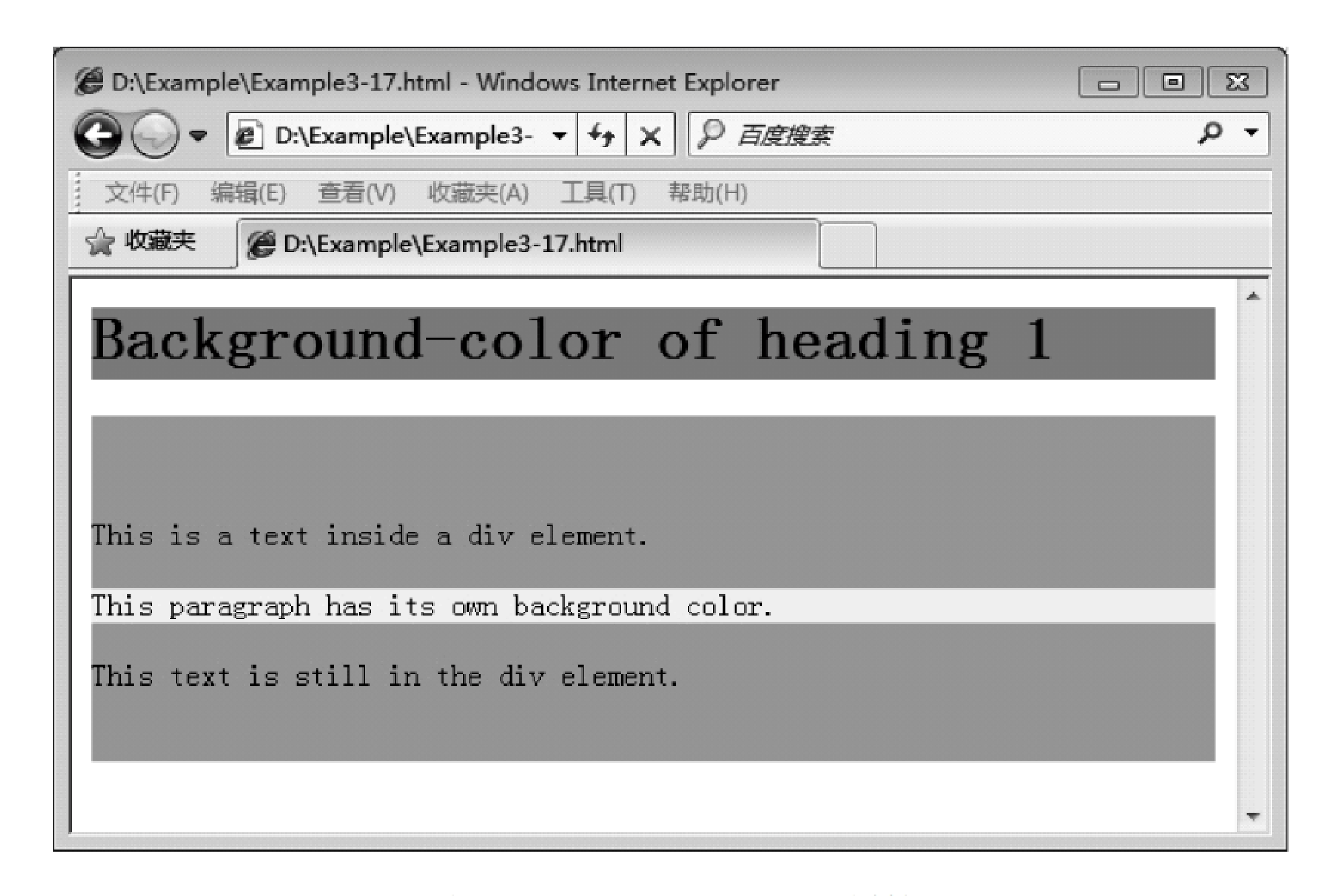


图 3.10 background-color 属性

2. 背景图片

属性：background-image

值：url(图片文件名称)

background-image 属性用来指定一张图片作为一个元素的背景。默认情况下，图片会不断重复以铺满整个元素。

整个页面的背景图片可以这样设置，如下。

例 3.18 设置背景图片。

```
<html>
  <head>
    <style type = "text/css">
    <! --
      body {background - image:url(wallpaper1.gif)}
    -->
    </style>
  </head>
  <body>
    <h1>Hello, World</h1>
  </body>
</html>
```

例 3.18 在浏览器中的显示效果如图 3.11 所示。



图 3.11 background-image 属性设置背景图片

3. 背景重复

属性：background-repeat

值：repeat（默认）| repeat-x | repeat-y | no-repeat

默认情况下，background-image 属性会同时横向和纵向重复，如例 3.18。有些图片仅仅需要横向或纵向重复，或者根本不需要重复。下面的代码表示图片 leaf.jpg 在浏览器窗口中只出现一次不重复：

```
body
{
background-image:url(leaf.jpg);
background-repeat:no-repeat;
}
```

4. 背景附着

属性：background-attachment

值：scroll（默认）| fixed

background-attachment 属性用来设置当下拉页面时背景图片是固定或者滚动。

例 3.19 设置背景图片不滚动不重复。

```
<html>
  <head>
    <style type="text/css">
    <!--
      body
      {
      background-image:url(leaf.jpg);
      background-repeat:no-repeat;
      background-attachment:fixed
      }
```

```
    -->
    </style>
  </head>
  <body>
    <h2>Chapter 1</h2>
    <p>This chapter discusses...</p>
    <br /><br /><br /><br /><br />
    <h2>Chapter 2</h2>
    <p>This chapter discusses...</p>
    <br /><br /><br /><br /><br />
    <h2>Chapter 3</h2>
    <p>This chapter discusses...</p>
    <br /><br /><br /><br /><br />
    <h2>Chapter 4</h2>
    <p>This chapter discusses...</p>
    <br /><br /><br /><br /><br />
    <h2>Chapter 5</h2>
    <p>This chapter discusses...</p>
    <br /><br /><br /><br /><br />
    <h2>Chapter 6</h2>
    <p>This chapter discusses...</p>
    <br /><br /><br /><br /><br />
  </body>
</html>
```

例 3.19 在浏览器中的显示效果如图 3.12 所示。

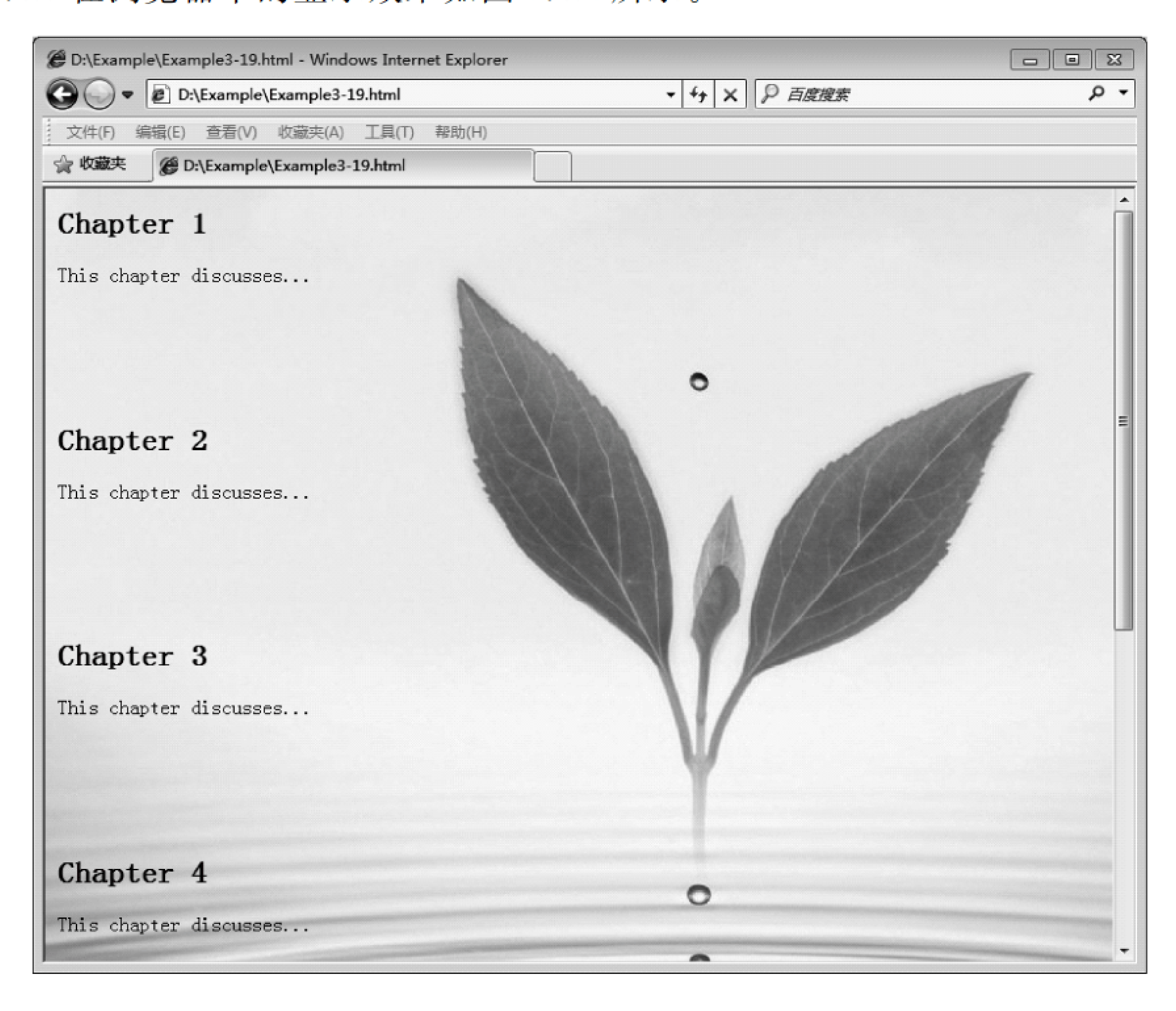


图 3.12　背景图片不滚动不重复

试着滚动例 3.19 中的网页，然后把第 9 行的属性值 fixed 改为 scroll，然后保存文件，刷新页面，再次滚动页面，改动后的网页将看起来很怪异。

5. 背景定位

属性：background-position

值：xpos ypos | x% y% | top left | top right | bottom left | …

background-position 属性指定背景图片的起始位置。例 3.19 中的样式表可以增加一个 background-position 属性如下。

例 3.20 改变图片默认定位。

```
body
{
background-image:url(leaf.jpg);
background-repeat:no-repeat;
background-attachment:fixed;
background-position: 20px 50px
}
```

当应用例 3.20 时，网页会看起来稍微有些奇怪。

6. 背景简写属性

属性：background

值(按顺序)：background-color background-image background-repeat background-attachment background-position

如前面学习过的字体属性一样，背景属性也可以使用简写办法在一个声明中设置所有属性。即使有一个或几个属性值没有给出也可以，只要有属性值的按上述顺序排列就可以。例 3.20 可以简写为：

```
body{background: url(leaf.jpg) no-repeat fixed 20px 50px}
```

提示：有了背景属性，设计者可以充分发挥自己的才能，把网页设计得华美绚丽。不过需要提醒读者注意的是，背景属性毕竟是背景，它是为前景的内容(文字、图片等)服务的，所以不要喧宾夺主，影响网站读者对内容的获取。如果是设置图片背景，首先要考虑的是小图片，由其水平及垂直方向重复，获得一个较好的背景图案，比如例 3.18；对类似例 3.19 的大图片背景使用要慎重，因为大图片文件大，会占用较多的带宽资源，导致网页下载速度慢。另外要注意的是，网页内容总是在背景前部的，所以如果图片本身是网页的内容，要使用 HTML 中的<img>标签来加入图片，要使背景图片与内容图片有所区别。

3.4.4 CSS 边框

CSS 边框属性定义元素周围的边框。

1. 边框样式

属性：border-style

值：none | dotted | dashed | solid | double | groove | ridge | inset | outset

border-style 属性用来设置边框的样式，默认情况下，边框的属性值是不显示(none)。

例 3.21 不同的边框样式。

```
<html>
  <head>
    <style type="text/css">
    <!--
      p.dotted {border-style:dotted}
      p.double {border-style:double}
      p.inset {border-style:inset}
      p.outset {border-style:outset}
     -->
    </style>
  </head>
  <body>
    <p class="dotted">A dotted border.</p>
    <p class="double">A double border.</p>
    <p class="inset">An inset border.</p>
    <p class="outset">An outset border.</p>
  </body>
</html>
```

例 3.21 在浏览器中的显示效果如图 3.13 所示。

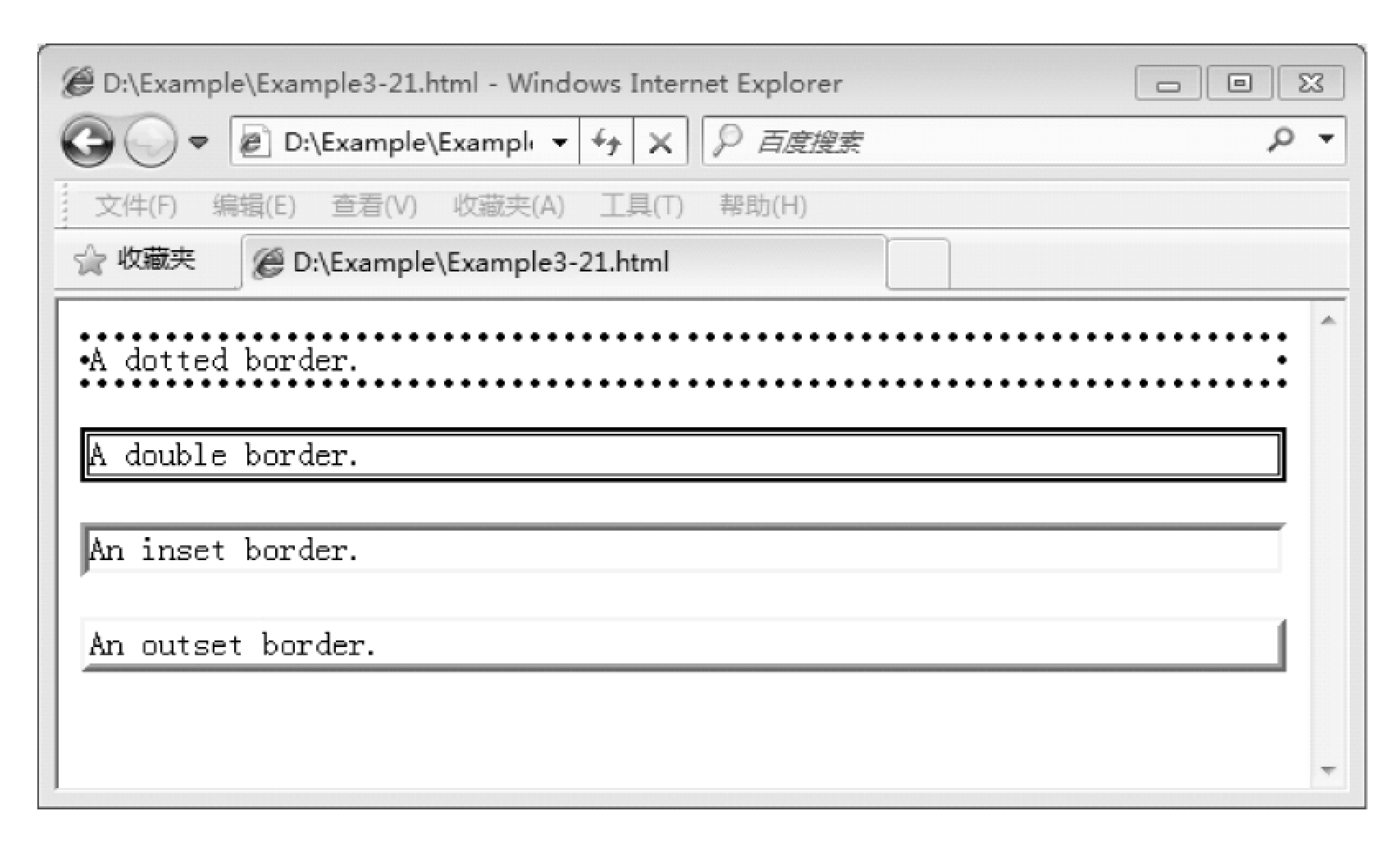


图 3.13 不同的边框样式

2. 边框宽度

属性：border-width

值：medium (默认) | thin | thick |长度(mm、px、em 等)

border-width 属性设置了边框的宽度。在设置 border-width 属性时需要先设置 border-style 属性。可以这样使用：

```
.border1 {border-style:solid; border-width:1em;}
```

3. 边框颜色

属性：border-color
值：颜色名称｜RGB｜十六进制
border-color 属性用来设置边框的颜色。可以这样使用：

```
.redborder {border-style:dashed; border-color:red;}
.blueborder{border-style:dashed;border-color:RGB(0,0,255);}
```

4. 边框简写属性

属性：border
值(按顺序)：border-width border-style border-color
在设置边框时有许多属性需要定义，为了简化代码，也可以在一个属性里定义所有的边框属性。可以这样使用：

```
.redborder {border: 1em dashed red}
```

提示：在边框的众多属性中，border-style 属性是必须设定的，如果这个属性没有设定，其他的属性设定都没有意义。另外，本小节的例子是设定元素四周的边框为相同的属性、粗细及颜色，边框还可以分别设定上、下、左、右边框为不同的风格、粗细和颜色，若想深入了解，可以参阅 CSS 详细资料。

3.4.5 CSS 外边距

CSS 外边距属性定义了元素周围的空间。外边距属性声明一个 HTML 元素和它周围元素之间的边距，它可以分别对元素的上、下、左、右的边距进行设置。

1. 外边距——单个边距

属性：margin-left、margin-right、margin-top、margin-bottom
值：%｜长度(in，mm，px，em，等)｜auto
当外边距(margin)的属性不是 auto 时，浏览器就会设置元素的边距，显示的效果视浏览器而定。可以这样使用：

```
.margin1
{
margin-top:10px;
margin-bottom:10px;
margin-right:30px;
margin-left:30px;
}
```

2. 外边距——简写属性

属性：margin

值(按顺序)：margin-top margin-right margin-bottom margin-left

元素的所有外边距也可以在一个属性里简写定义，如果 4 个属性值都被定义，那么顺序是：top(上)、right(右)、bottom(下)、left(左)，例如：

```
.margin1 {margin: 10px 30px 10px 30px}
```

如果只定义一个值，那么它将设置所有的外边距，例如：

```
.margin1 {margin: 10px }
```

如果只定义两个或三个值，没有定义的外边距将采用对边的外边距，例如：

```
.margin1 {margin: 10px 30px }              /* 定义两个值 */
.margin2 {margin: 10px 30px 10px}          /* 定义三个值 */
```

例 3.22 不同设置的外边距。

```
<html>
  <head>
    <title>Example of CSS border and margin</title>
    <style type="text/css">
    <!--
      .d1{border:2px solid #ff0000}
      .d2{border:2px solid gray}
      .d3{margin:10px 30px;border:2px solid gray}
    -->
    </style>
  </head>
  <body>
    <div class="d1">
      <div class="d2">No margin declared</div>
    </div><br /><br /><br />
    <div class="d1">
      <div class="d3">Two margin values are declared</div>
    </div>
  </body>
</html>
```

例 3.22 在浏览器中的显示效果如图 3.14 所示。注意在两种情况下，内部灰色边框(2px solid gray)和外部红色边框(2px solid red)之间的外边距。

提示：外边距是排版常用的概念，在字处理软件中，也有页边距的概念。在网页中，一个元素和另外一个元素之间有上、下、左、右四个外边距，这四个外边距的设置可以通过四个单独的外边距属性设置，但从上边的讨论中可以看出，这种设置办法书写麻烦，用快捷外边距属性 margin 即可一次完成四个外边距的定义。从 margin 的使用讨论中可知，如果四个外边距值相同，给出一个值即可；如果上下外边距相同，左右外边距相同，设置两个值即可；如果设置的是不对称的外边距，就需要设置三个以上的值。

例 3.22 通过外部的红色边框和内部的灰色边框，说明了不定义外边距和定义外边距的区别。

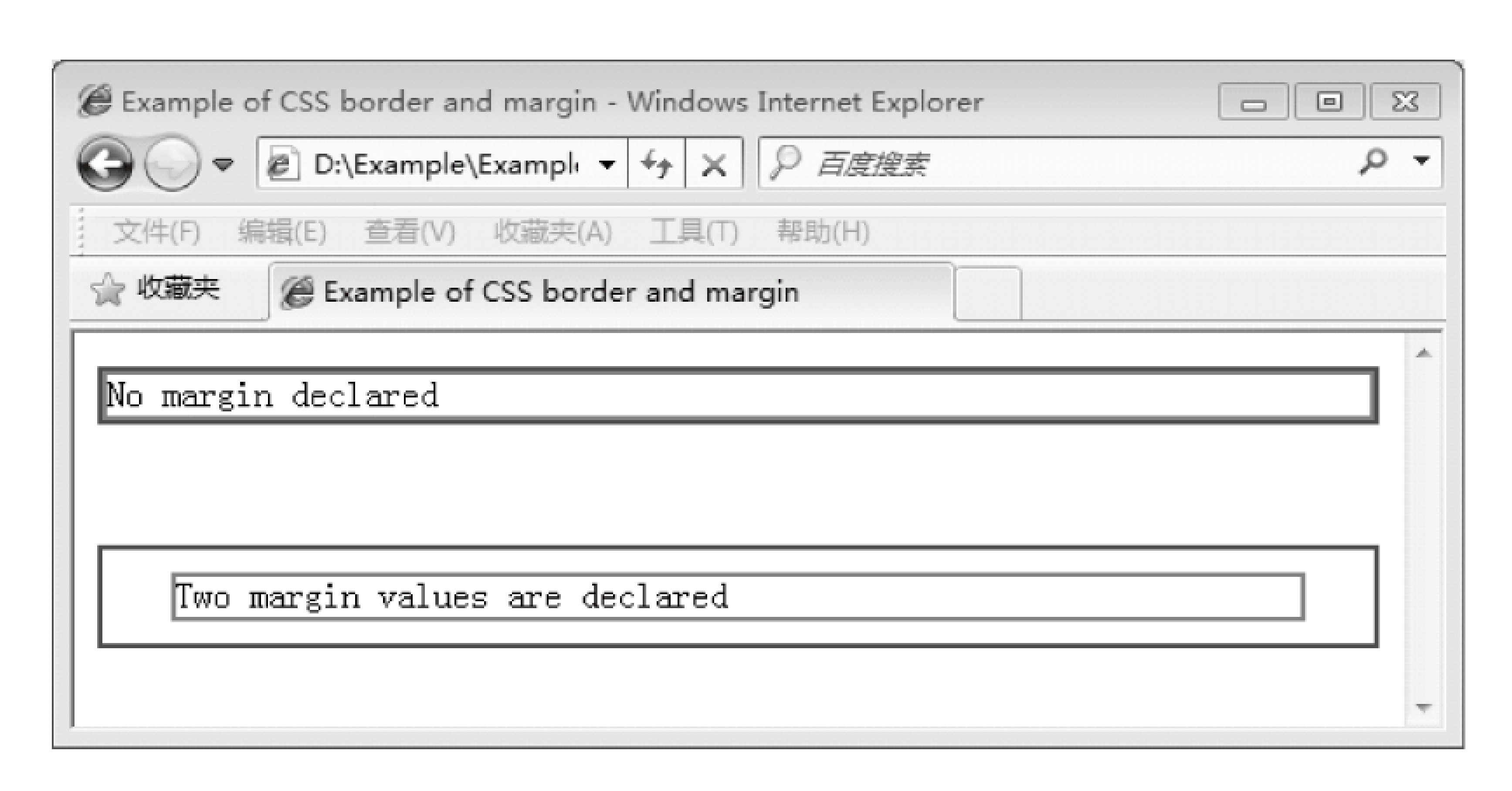


图 3.14 边框和外边距示例

3.4.6 CSS 内边距

CSS 内边距定义的是一个 HTML 元素边框和它里面内容之间的距离。除了内边距属性值中没有 auto，许多外边距的规则也同样适用于内边距。

1. 内边距——单个边距

属性：padding-left、padding-right、padding-top、padding-bottom

值：% |长度(in、mm、px、em 等)

可以为不同的边设置不同的内边距，如下所示：

```
.padding1
{
padding-top:10px;
padding-bottom:10px;
padding-right:30px;
padding-left:30px;
}
```

2. 内边距——简写属性

属性：padding

值(按顺序)：padding-top padding-right padding-bottom padding-left

元素的所有内边距也可以在一个属性里定义，如果 4 个属性值都被定义，那么顺序是：top(上)、right(右)、bottom(下)、left(左)，例如：

```
.padding1 {padding: 10px 30px 10px 30px}
```

如果只定义一个值，那么它将设置所有的内边距，例如：

```
.padding1 {padding: 10px }
```

如果只定义两个或三个值，没有定义的内边距将采用对边的内边距，例如：

```
.padding1 {padding: 10px 30px }              /* 定义两个值 */
.padding2 {padding: 10px 30px 10px}          /* 定义三个值 */
```

下面的例子显示了外边距和内边距之间的不同。

例 3.23 外边距与内边距的分别。

```
<html>
  <head>
    <title>Difference of margin and padding</title>
    <style type="text/css">
    <!--
      .d1{border:2px solid #ff0000}
      .d2{border:2px solid gray}
      .d3{margin:10px 30px;border:2px solid gray}
      .p1{padding:10px 30px}
    -->
    </style>
  </head>
  <body>
    <div class="d1">
      <div class="d2 p1">No margin declared. Two padding values are declared.</div>
    </div><br /><br /><br />
    <div class="d1">
      <div class="d3">Two margin values are declared. No padding declared.</div>
    </div><br /><br />
    <div class="d1">
      <div class="d3 p1">Two margin values and two padding values are declared. </div>
    </div>
  </body>
</html>
```

例 3.23 在浏览器中的显示效果如图 3.15 所示。

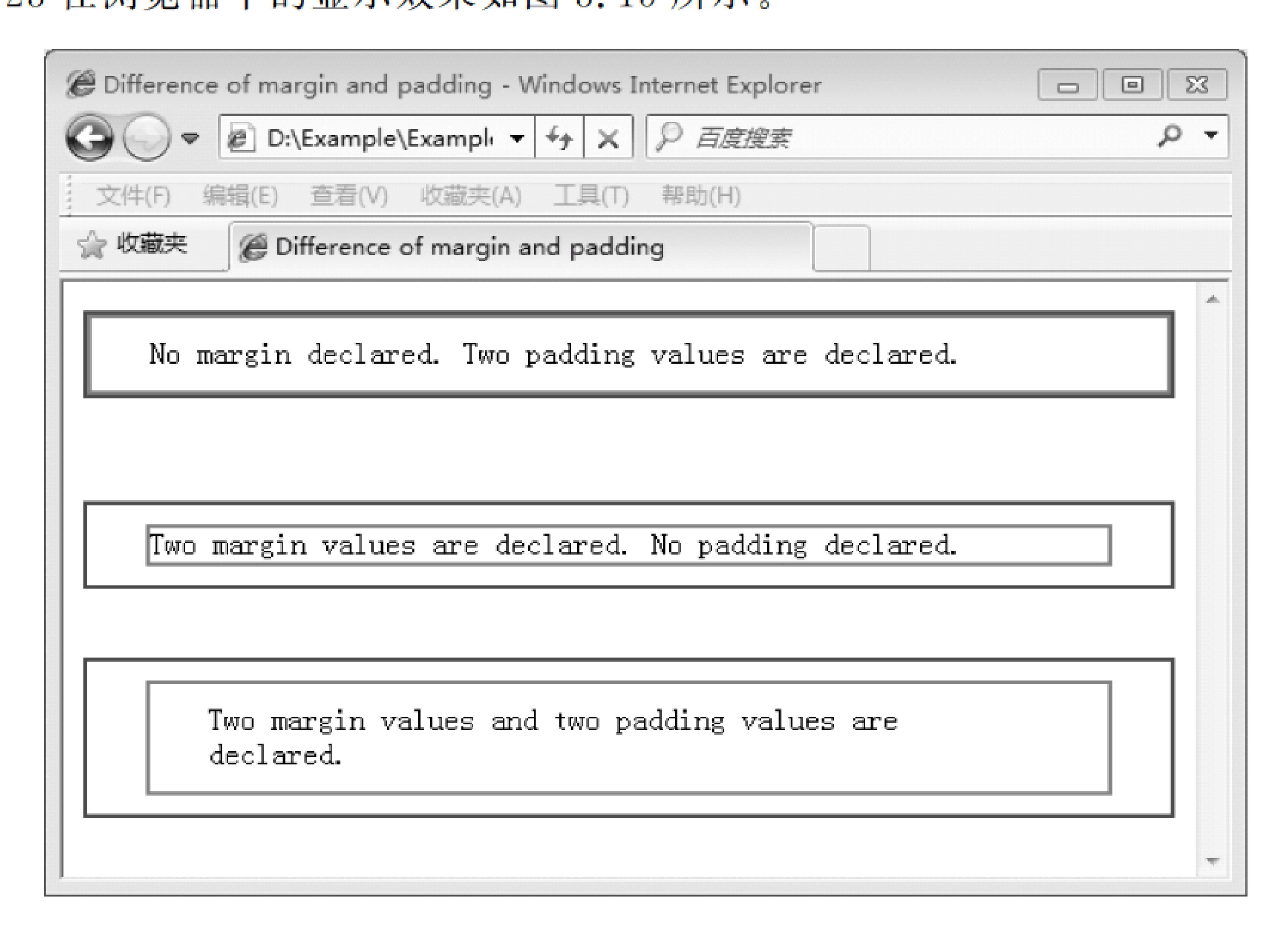


图 3.15 外边距和内边距之间的不同

提示：内边距属性(padding)与外边距属性(margin)这两种属性在使用时十分相似，但应用时也十分容易混淆。因此以例 3.23 来说明这种区别：这里首先引用了边框，因为边框实际就是元素的边界的显性显示，例子中以灰色边框作为文字内容的边界。内边距属性是用来设置元素内容到元素边界的距离，所以从图 3.15 中间的边框示意可以看到没有声明内边距时，灰色边框是紧贴着里边的文字(即内容)，最上边和最下边的两个边框都是声明了内边距的情况，文字(内容)和灰色边框之间就存在内边距；外边距属性是用来设置一个元素所占空间的边缘到相邻元素之间的距离，图 3.15 最上边的灰色边框与红色边框是没有声明外边距的情况，而中间和下边的两个是声明外边距的情况。

为了更详细地理解，可以参见 3.4.7 节“CSS 盒模型”。

3.4.7 CSS 盒模型

网页设计中的每个元素都是一个矩形盒。在 CSS 中，进行设计和布局时，会用到盒模型。CSS 盒模型从本质上来说是环绕在 HTML 元素周围的方框，它由外边距、边框、内边距和实际内容组成。

盒模型允许放置元素周围的边框及周围相关元素，图 3.16 阐释了盒模型。

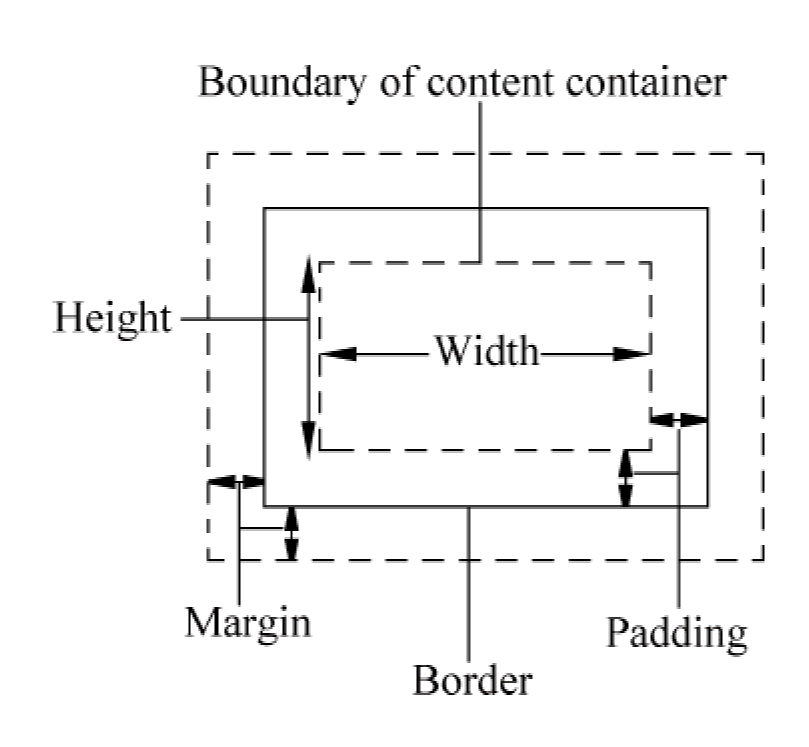


图 3.16 CSS 盒模型

为了能在所有的浏览器中正确地设置元素的宽(width)和高(height)，需要了解盒模型的工作原理。当一个元素的宽和高被 CSS 定义后，它设置的仅仅是内容区域的宽度和高度。元素(盒)的总宽度还包括内边距(padding)、边框(border)和外边距(margin)，如下例子中元素的总宽度为 300px。

```
.element1
{
width:250px;
padding:10px;
border:5px solid gray;
margin:10px;
}
```

计算过程如下：

250px(宽度)+ 20px(左右内边距)+ 10px(左右边框)+ 20px(左右外边距)= 300px

元素(盒模型)的总宽度与总高度应当这样计算：

```
盒的总宽度 = width + padding-left + padding-right + border-left + border-right +
margin-left + margin-right
盒的总高度 = height + padding-top + padding-bottom + border-top + border-bottom +
margin-top + margin-bottom
```

图 3.17 阐释了盒模型的层次关系。

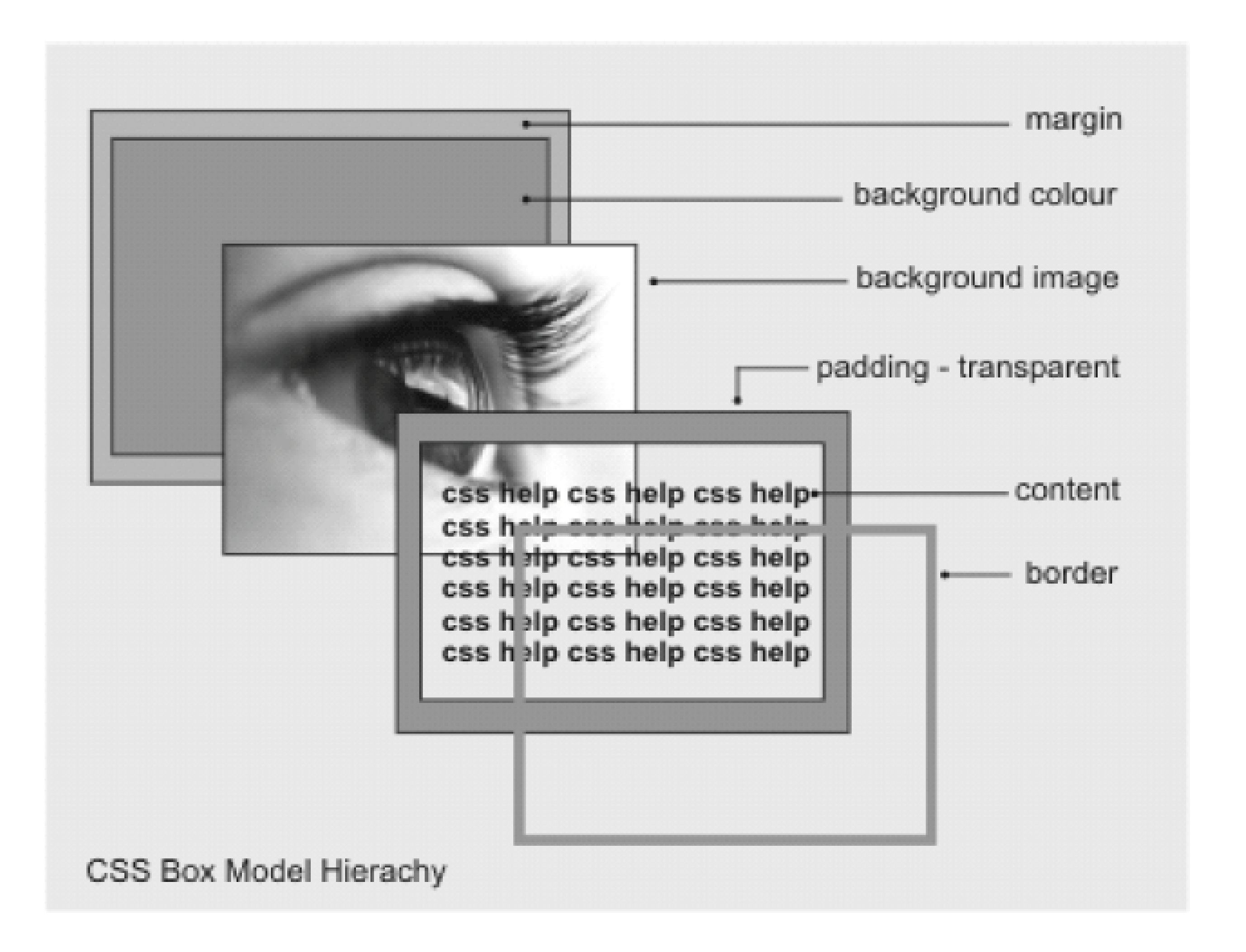


图 3.17 CSS 盒模型层次

例 3.24 盒模型中各个属性概念。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 1.0 Transitional//EN"
"http://www.w3.org/TR/HTML1/DTD/HTML1-transitional.dtd">
<html>
  <head>
    <title>Example of Box Model</title>
    <style type="text/css">
    <!--
      .boxmodel
      {
      width: 280px;
      height: 280px;
      background-image: url(puppy.jpg);
      }
     -->
    </style>
  </head>
  <body>
    <p class="boxmodel">This is the content of a paragraph. The width of the paragraph has
been defined.</p>
  </body>
</html>
```

例 3.24 在浏览器中的显示效果如图 3.18 所示。

下面开始为例 3.24 添加不同的盒模型元素。

设置内边距(padding)为 20px,如下:

```
.boxmodel
{
width: 280px;
height: 280px;
background-image: url(puppy.jpg);
```

```
padding: 20px;
}
```

修改后的例 3.24 在浏览器中显示效果如图 3.19 所示。

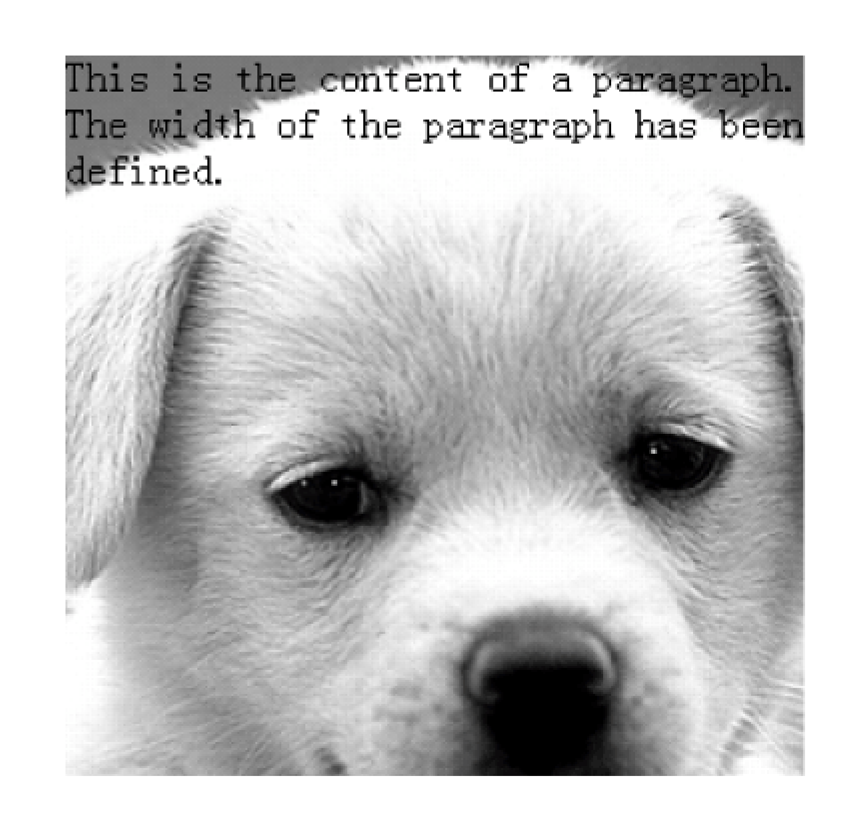


图 3.18 一个有背景图片的段落

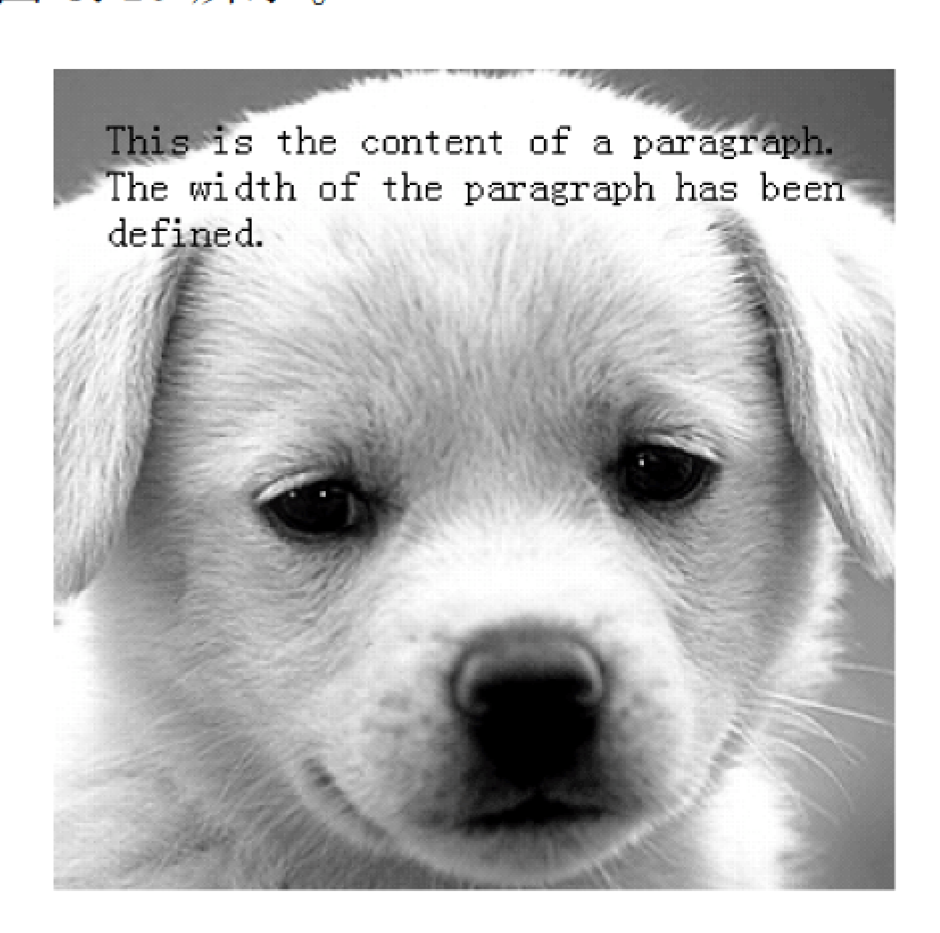


图 3.19 有 20px 内边距的段落

设置边框为 20px，灰色，如下：

```
.boxmodel
{
width: 280px;
height: 280px;
background-image: url(puppy.jpg);
border: solid gray 20px;
}
```

本次修改后的例 3.24 在浏览器中显示效果如图 3.20 所示。

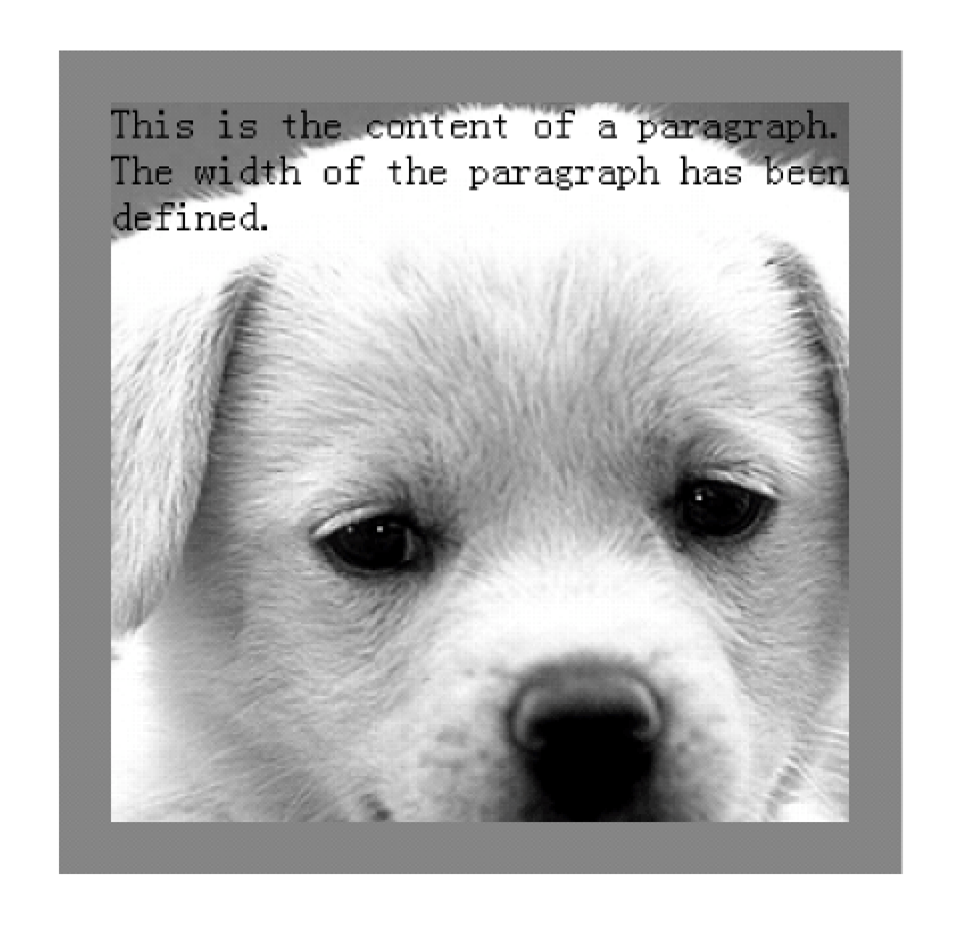


图 3.20 有 20px 灰色边框的段落

设置外边距为 20px，如下：

```
.boxmodel
{
width: 280px;
height: 280px;
```

```
background-image: url(puppy.jpg);
margin: 20px;
}
```

本次修改后的例 3.24 在浏览器中显示效果如图 3.21 所示。

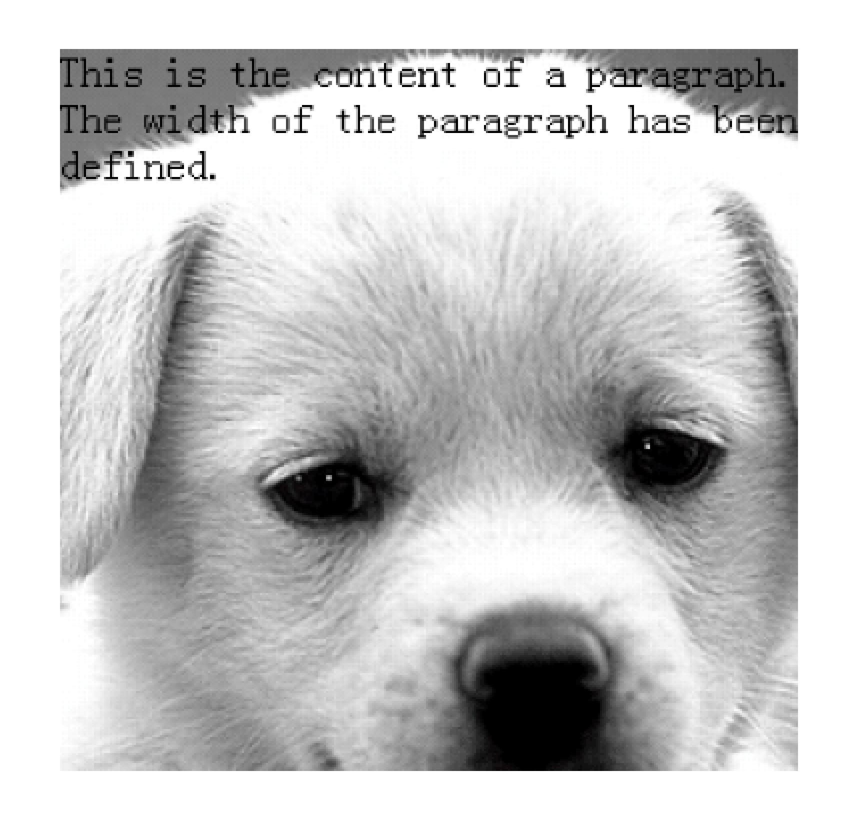


图 3.21　有 20px 外边距的段落

说明：图 3.22 的 20px 外边距，由于周围没有其他元素，图片中无法看出效果。对比图 3.23 中的 20px 外边距，灰色边框与浏览器上、左边界之间有距离，空白距离就是外边距。

设置所有的盒模型元素如下：

```
.boxmodel
{
width: 280px;
height: 280px;
background-image: url(puppy.jpg);
margin: 20px; padding: 20px; border: solid gray 20px;
}
```

最终，修改后的例 3.24 在浏览器中的显示效果如图 3.22 所示。

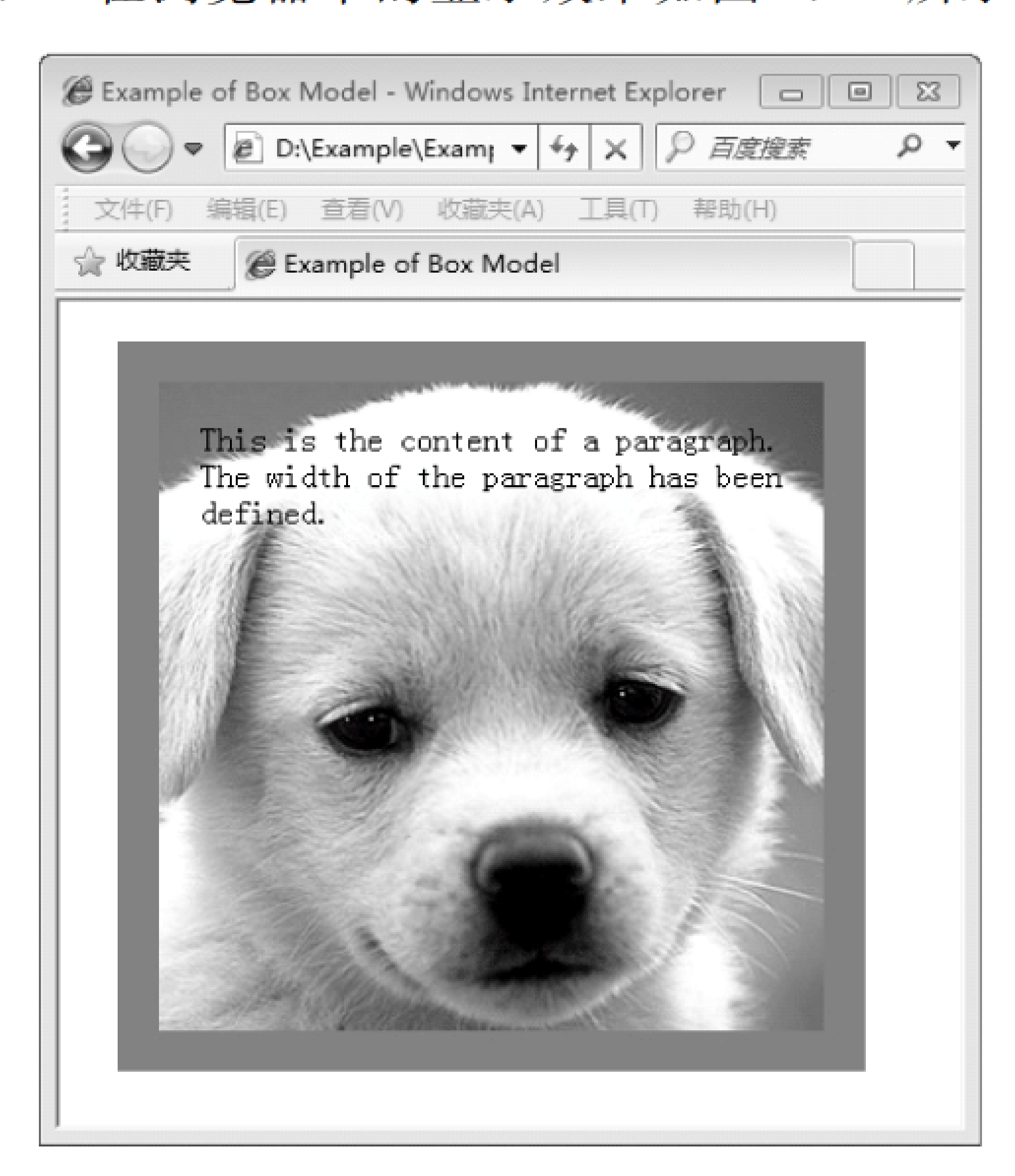


图 3.22　加入了所有盒模型元素

在沃克俱乐部主页(index.html)中使用所有盒模型元素，如下项目3.3和项目3.4所示。

项目3.3　CSS文件layout.css。

```
body
{
background-color:rgb(0,204,153);
font-size:medium;
font-family:宋体;
margin:0;
}
h1,h3{font-family:黑体;}
h1{font-size:200%;color:red;}
h3{font-size:120%;color:green}
div.content{text-indent:1.5em}          /* 设定首行缩进 */
.center{text-align:center}              /* 设定文本居中对齐 */
hr{width:80%}                           /* 设定横线 */

/*定义网页页头id,并放入背景图片*/
#header
{
/*背景图片banner.jpg宽800px,高224px*/
width:800px;
height:224px;
margin:0 auto;
background-image:url(images/banner.jpg);
}

/*定义网页正文内容容器,在网页中只引用一次*/
#container
{
/*本容器上外边距设定为0,和上部图片有紧密结合,按照盒模型概念,这里定义的元素宽度、外边
距和内边距合计宽度为800px,高度不进行定义,由正文内容自动生成*/
width:784px;
margin:0 auto;
padding:8px;
background-color:rgb(187,224,227);
}

/*定义网页页脚id*/
#footer{
width:784px;
margin:0 auto;
padding:8px;
background-color:rgb(187,224,227);
}
```

项目3.4　主页文件index.html。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 1.0 Transitional//EN" "http://www.w3.org/TR/HTML1/
DTD/HTML1-transitional.dtd">
```

```
<html>
  <head xmlns="http://www.w3.org/1999/HTML">
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
    <title>沃克俱乐部欢迎您</title>
    <!-- 链接外部样式表文件 layout.css -->
    <link rel="stylesheet" type="text/css" href="layout.css" />
  </head>
  <body>
    <div id="header" class="center"><!-- 调用页头 id 及居中类 -->
    </div>
    <div  id="container"><!-- 调用正文容器 id -->
      <div class="center"><!-- 调用文本居中对齐类 -->
        <p>
        <a href="introduction/article1.html" target="_blank">健走的方法</a> 
  
        <a href="introduction/article2.html" target="_blank">健走装备</a> 
  
        <a href="activity/activity1.html" target="_blank">俱乐部活动</a>
        </p>
      </div>
      <h3>走路,简单有效的健身之道</h3>
      <div class="content">  <!-- 调用首行缩进类 -->
        <p>走路是人类本能的身体活动,人类身体结构就是为步行设计的。问题是,随着汽车等代
步工具的普及,人们"好逸恶劳"的条件和理由越来越充足,以致基本的走路功能都有些退化了。根据
美国疾病控制与预防中心的研究,有高达 75% 的美国人每天运动量不到 30 分钟,大部分人几乎成天
坐在椅子上,是不折不扣的"沙发土豆"。反观我们中国的现实情形,随着经济的发展,汽车的大量使
用及其他因素,导致缺乏锻炼、体形肥胖的人越来越多。</p>
        <p>走路健身又健脑。我国自古就有"走为百练之祖"的健身经验谈,其中传统医学认为,双
脚是人体的健康之根。走路刺激脚底穴位,能舒筋通络,活血顺气,强身健体。同时,现代运动医学研
究也证实: 走路时,骨骼、肌肉、韧带、神经末梢都要参加运动,从而促进血液循环,调节大脑皮层的活
动功能,促使身体各种激素分泌,使人心情愉悦。走路防病并延缓衰老。最新的医学研究表明,一周
健步走 7 小时以上,可以降低 20% 乳腺癌、30% 心脏病和 50% 糖尿病的罹患率,而中老年人每天散步
2.4 公里以上,心脏病发作率将降低 50%。</p>
      </div>
    </div>
    <div id="footer" class="center"><!-- 调用页脚 id 及居中类 -->
      <hr />
      <p><a href="mailto:webmaster@walkerclub.net">和我联系 Email: webmaster@
walkerclub.net</a></p>
    </div>
  </body>
</html>
```

项目 3.4 在浏览器中的显示效果如图 3.23 所示。

提示:在学习了 CSS 的边框、外边距、内边距等属性之后,学习盒模型是对这些属性的汇总。只有了解盒模型,才可以综合应用这个概念进行网页的设计。本小节的项目 3.3 和项目 3.4 对这盒模型的概念做了一个应用,这种设计已经接近于实际应用的网页。在熟练掌握这些概念之后,可以尝试在正文内容容器(layout.css 中的#container)之内再定义容器即盒模型,添加其他内容,或者对网页顶部图片进行处理(比如加入文字、制作公司 logo、



图 3.23 应用了 CSS 属性的主页

制作成动画文件等)，可以使网页逐步完美。

3.4.8 CSS 伪类

伪类与类相似，但不是真正的类，伪类用来增加一些特殊效果。

伪类语法：

```
selector:pseudo - class {property:value}
```

伪类以冒号(:)开始，可以这样使用：

```
a:link{color:red}
```

链接(<a>标签)伪类是最常见的伪类。有 4 种链接伪类如下：

:link 指定未访问的链接样式。

:visited 指定已经访问过的链接样式。

:hover 指定当鼠标悬浮时的链接样式。

:active 指定当被单击时的链接样式。

例 3.25 设定链接样式。

```
< html >
  < head >
```

```
    <style type="text/css">
    <!--
      a:link {color:red}                  /* 未访问时 */
      a:visited {color:green}             /* 已访问过 */
      a:hover {color:orange}              /* 鼠标悬停时 */
      a:active {color:blue}               /* 鼠标点中时 */
    -->
    </style>
  </head>
  <body>
    <p><a href="Example3.24.html" target="_blank">This is a link specified anchor pseudo
-classes.</a></p>
  </body>
</html>
```

在浏览器中证实以上例子中链接在鼠标悬浮和单击时的颜色。

注意：在CSS定义时，a:hover必须跟在a:link和a:visited之后，a:active必须跟在a:hover之后。

提示：伪类和类相似，但不是真正的类。从本小节中的语法上可以看出相似度很大。伪类可以实现类所实现不了的一些特效。W3C制订的CSS标准，从CSS1到CSS2和CSS3，一直在不断地添加一些新的伪类，以丰富CSS的表现能力，只是这些新添加的许多伪类尚未被多数主流浏览器所支持，所以应用不多。伪类是CSS的一个发展方向，可以多加关注，但要注意的是使用的伪类要被所有的主流浏览器支持，否则会影响使用。

本小节所述的链接伪类，已经被所有主流浏览器所支持，可以放心使用。

要看到例3.25中链接中四种不同情况下的四种颜色，要注意在浏览器中清除历史记录，这样首先保证未访问链接(a:link)的颜色可以正常显示，悬浮(a:hover)和单击(a:active)两个颜色不会有问题，对访问过的链接，必须保证链接是一个真实链接而非空链接，这样访问过的链接(a:visited)会被记录到浏览器中，这样已访问链接颜色才可以正确显示。

3.5 CSS3

CSS3是层叠样式表的最新标准。CSS3基于目前使用的CSS2标准的基础上，添加了新的特性。

使用CSS3，可以使得设计人员只用样式表的办法就可以对网页元素完成高级的样式设计。比如在CSS2中，元素边框是方形样式，如果要设计成为圆角样式，必须借助图片的方式进行复杂而繁琐的处理才能够完成。利用CSS3，仅需简单指定边框的属性，就可以完成圆角边框的设计。

由于CSS3的标准仍然在制订过程中，为使制订过程简化，整个CSS3标准被分为一个个模块，每个模块对应特定的设计与特性。如颜色模块、选择符模块、字体模块、边框模块等。

3.5.1 CSS3 边框

1. 设置圆形边框

前页谈到，使用 CSS3 可以完成圆形边框的设计。参见例 3.26。

例 3.26 圆形边框。

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    border: 2px solid #a1a1a1;
    padding: 10px 40px;
    background: #dddddd;
    width: 300px;
    border - radius: 25px;
}
</style>
</head>
<body>
<div>The border - radius property create a rounded box.</div>

</body>
</html>
```

效果如图 3.24 所示。

The border-radius property create a rounded box.

图 3.24 CSS3 圆形边框

圆形边框语法(border-radius)：

```
border - radius:1 - 4 length value
border - * - radius:length value
```

说明：边框半径值可以选取 1 到 4 个长度值。类似于边框(border)的取值方法，4 个值按顺序依次对应左上角、右上角、右下角和左下角。如果只有一个值，则四个角取相同值；如果只有两个值，则说明是左上角、右上角取值，按对称原则，右下角的值与左上角相同，左下角的值与右上角相同；如果取三个值，没有取值的最后一个是左下角，其值与右上角相等。

如：

```
border - radius:25px 10px;
```

实际上与如下的写法有相同效果：

```
border - top - left - radius:25px;
```

```
border - top - right - radius:10px;
border - bottom - right - radius:25px;
border - bottom - left - radius:10px;
```

2. 盒子添加阴影

看如下的例子。

例 3.27 盒子阴影。

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width: 300px;
    height: 100px;
    background - color: yellow;
    box - shadow: 10px 10px 5px #888888;
}
</style>
</head>
<body>
<div></div>
</body>
</html>
```

例 3.27 在浏览器产生的效果如图 3.25 所示。

图 3.25 带阴影的盒子

阴影语法：

```
box - shadow: h - shadow v - shadow blur spread color
```

说明：

box-shadow 向盒子(框)添加阴影。该属性是由逗号分隔的阴影列表，每个阴影由 2～4 个长度值、可选的颜色值来规定。

h-shadow：必需。水平阴影位置，允许负值(正值向右投影，负值向左投影)。

v-shadow：必需。垂直阴影位置，允许负值(正值向下投影，负值向上投影)。

blur：可选。模糊距离。

spread：可选。阴影尺寸。

color：可选。阴影颜色，缺省为黑色。

盒子阴影支持 IE9 及以上版本，Chrome，Firefox。

3.5.2 CSS3 渐变

CSS3 渐变可以显示出指定两种或多种颜色的平滑过渡。

在 CSS3 出现之前，渐变效果是通过图片处理软件（如 Photoshop）处理的图片得到。使用 CSS3 完成渐变，用户可以减少下载图片的时间和带宽使用。另外，使用 CSS3 完成的渐变在网页缩放时有更好的视觉效果，因为 CSS3 生成的渐变不是图片，而是通过浏览器生成。

CSS3 定义两种渐变：

线性渐变（向上，向下，向左，向右，斜向）；

径向渐变（圆形，椭圆形）。

1. 线性渐变

首先看下面的例子。

例 3.28 线性渐变效果。

```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
    height: 200px;
    width: 600px;
    background: linear-gradient(to right, red , blue);
}
#grad2 {
    height: 200px;
    width: 600px;
    background: linear-gradient(to bottom right, red , blue);
}
#grad3 {
    height: 200px;
    width: 600px;
    background: linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1));
}
</style>
</head>
<body>
    <h3>Linear Gradient</h3>
    <p>This linear gradient starts at the left. It starts red, transitioning to blue:</p>
    <div id="grad1"></div>
    <p>This linear gradient starts at the top left and goes to bottom right. It starts red,
transitioning to blue:</p>
    <div id="grad2"></div>
    <p>This linear gradient starts at the left. It use the rgba() function to define
transparency:</p>
    <div id="grad3"></div>
```

```
    <p><strong>Note:</strong>IE10+, Chrome 26+ and Firefox 16+ support gradients.</p>
</body>
</html>
```

由于本书的灰度图像无法准确显示出色彩的渐变效果，因此，本例不进行截图显示。读者可以使用IE10以上版本，或者最新的Chrome、Firefox直接在电脑上查看本例的渐变效果图。

创建线性渐变，必须定义至少两种颜色，即起始颜色与结束颜色，渐变就在这两个颜色之间平滑渐变。渐变方向可以设定。

线性渐变语法：

```
background: linear-gradient(direction, color-stop1, color-stop2,...)
```

说明：

direction：线性渐变方向，to bottom（缺省）、to top、to left、to right。

color-stop1：起始颜色。

color-stop2：结束颜色（如果有第3个颜色，它又是下一个线性渐变的起始颜色）。

例3.28中的选择符#grad1（第1个渐变）显示了从左向右的渐变，以红色开始，渐变到蓝色。

例3.28中的选择符#grad2（第2个渐变）显示了从左上角向右下角的渐变，以红色开始，渐变到蓝色。

CSS3渐变还支持透明效果，即产生一种淡入淡出的效果。

完成透明效果，需使用rgba()函数定义起始颜色。rgba()函数的最后一个参数定义了颜色的透明度，取值范围从0到1，0是全透明，1是全色（不透明）。

例3.28中的选择符#grad3（第3个渐变）显示了从左向右的透明效果，以透明开始，到完全的红色。

2. 径向渐变

首先看下面例子。

例3.29 径向渐变效果。

```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
    height: 150px;
    width: 200px;
    background: radial-gradient(red, yellow, blue);
}
#grad2 {
    height: 150px;
    width: 200px;
    background: radial-gradient(circle, red, yellow, blue);
}
```

```
</style>
</head>
<body>
    <h3>Radial Gradient - Shapes</h3>
    <p>Ellipse (this is default):</p>
    <div id="grad1"></div>
    <p>Circle:</p>
    <div id="grad2"></div>
    <p><strong>Note:</strong> IE10+, Chrome 26+ and Firefox 16+ support gradients.</p>
</body>
</html>
```

由于本书的灰度图像无法准确显示出渐变效果，因此，这里不对例 3.29 运行效果截图显示。读者可以使用 IE10 以上版本，或者最新的 Chrome、Firefox 直接在电脑上查看本例的渐变效果图。

径向渐变需要定义一个圆心点。然后定义渐变需要的至少两种颜色、形状(圆形或椭圆形)及尺寸。缺省情况下，圆心为几何中心，形状是椭圆，尺寸到当前元素的最远端。

径向渐变语法：

```
background: radial-gradient(center, shape size, start-color, ..., last-color)
```

说明： Shape 参数定义形状，取值为 circle 或 ellipse(缺省值)。

例 3.29 中选择符 #grad1(第一个渐变效果)无 shape 参数，因此渐变的形状为缺省的椭圆。例 3.29 中选择符 #grad2(第二个渐变效果)指定 shape 参数为 circle，渐变形状为圆。

提示： 渐变有许多参数与使用方法，有兴趣的读者可以参考 CSS3 详细用法与技巧。

3.5.3 CSS3 文本效果

CSS3 在文本显示效果上添加了一些新的特性。截止本书完成之时，主流浏览器对多数 CSS3 文本效果属性支持情况并不好。本小节中，只介绍一下文本效果中的文本阴影属性，本属性被主流浏览器所支持。

例 3.30 文本阴影效果。

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
    text-shadow: 5px 5px 5px #FF0000;
}
</style>
</head>
<body>
    <h1>Text-shadow effect!</h1>
</body>
```

```
</html>
```

例 3.30 在浏览器的显示效果如图 3.26 所示。

Text-shadow effect!

图 3.26 文本阴影属性

文本阴影属性语法：

```
text - shadow: h - shadow v - shadow blur color
```

说明：

h-shadow：必需。水平阴影位置，允许负值(正值向右投影，负值向左投影)。
v-shadow：必需。垂直阴影位置，允许负值(正值向下投影，负值向上投影)。
blur：可选。模糊距离。
color：可选。阴影颜色。
IE9 及以下版本不支持文本阴影属性。

小结

内容与形式分离是当今网页设计的思想。内容、结构由 HTML 负责，比如标题、段落、图片、链接及 HTML5 新引入的网页布局元素等；形式、外观由 CSS 负责，比如字体、背景、边框及 CSS3 引入的渐变效果等。使用时两者结合，各负其责，能够高效率地设计与维护网页。这种开发方式实际上就是当前的 Web 设计标准 DIV ＋ CSS。

本章自始至终都在按 DIV ＋ CSS 的标准进行讲授。在完成本章的学习之后，第 2 章使用的行内样式表(也称内联样式表)不再推荐使用，内部样式表也尽量少用，强烈推荐符合 DIV ＋ CSS 的外部样式表完成网页的设计。

本章最后部分简要介绍了 CSS 最新版本 CSS3 一些特性。利用 CSS3 属性，可以轻松完成 CSS 以往版本上实现起来非常麻烦、非常高级的设计。需要指出的是，虽然 CSS3 实现了多种炫酷的网页设计效果，但要注意浏览器对 CSS3 的支持情况，尤其是 IE，一般都需要 IE10 以上才可以实现 CSS3 的设计效果，因此，在使用 CSS3 进行设计时，需要加入对老版本浏览器的兼容性代码。CSS3 仍然在发展中，本书只是对 CSS3 做一个指导性的介绍，CSS3 是发展的方向，读者在实际设计中，可以关注 CSS3 的应用。

习题

1. CSS 利用哪个 HTML 标签进行网页布局？
   A. <dir>　　B. <div>　　C. <dis>　　D. <dif>
2. 下列哪一项是 CSS 的正确语法？
   A. body. background-color = rgb(22,38,126)；
   B. body：background-color = { rgb(22,38,126)}；

C. body{background-color：rgb(22,38,126);}

D. {body background-color = "rgb(22,38,126)";}

3. 如何给所有的<h3>标签设置字体颜色？

A. .h3{color:green}　　B. h3{color:green}

C. all h3{color:green}　　D. #h3{color:green}

4. 下列哪一个是 HTML 文件引用外部样式表文件的正确语句？

A. <style src="layout.css">

B. <stylesheet>layout.css</stylesheet>

C. <import @layout.css></import>

D. <link rel="stylesheet" type="text/css" href="layout.css" />

5. 如何去掉超链接默认的下划线？

6. 类.element1 的上外边距与下外边距相同，都是 20px，左外边距与右外边距相同，都是 5px。使用外边距的简写方式，如果完成？

7. 设计一个搜索文本框，要求文本框为一个圆角文本框，需要利用 CSS3 的哪一个属性完成？

第4章 jQuery

jQuery 是一套跨浏览器的、轻量级的 JavaScript 库，简化 HTML 与 JavaScript 之间的操作。相对于 jQuery，JavaScript 语言是基础，jQuery 是 JavaScript 语言众多库中的一个（此外还有 Prototype、Dojo、YUI、Ext JS 等）；但无疑 jQuery 是 JavaScript 语言众多库中最闪耀的一个，如没有 jQuery 的锦上添花，JavaScript 语言会减色不少。

jQuery 由美国人 John Resig 在 2006 年 1 月上发布第一个版本，目前是由 Dave Methvin 领导的开发团队进行开发，是目前最受欢迎的 JavaScript 库。jQuery 的宗旨是"write less, do more"（写的更少，做的更多）。

jQuery 是开源软件，使用麻省理工学院的 MIT 许可证授权。jQuery 的语法设计使得许多操作变得容易，如操作文档对象、选择 DOM 元素、创建动画效果、处理事件以及开发 Ajax 程序。jQuery 也提供了给开发人员在其上创建插件的能力。这使开发人员可以对底层交互与动画、高级效果和高级主题化的组件进行抽象化。模块化的方式使 jQuery 函数库能够创建功能强大的动态网页以及网络应用程序。

4.1 JavaScript 基础

JavaScript 语言是目前 Internet 上最流行的、最重要的编程语言之一，它被无数的网页用作改进设计、验证表单、检测浏览器，以及更多的应用，学习 Web 技术必须学会 JavaScript 语言。JavaScript 语言可以使页面更加生动、灵活，可以用极小的程序量完成需要的功能。JavaScript 语言的基本结构形式与 C、C++、VB、Delphi 等十分类似，但它不像这些语言一样，需要先编译，JavaScript 语言是一种脚本语言，在程序运行过程时被逐行地解释运行。JavaScript 语言可以与 HTML 语言紧密结合在一起，从而方便用户的使用操作。

JavaScript 语言的出现，使得信息和用户之间不仅只是一种显示和浏览的关系，而是实现了一种实时的、动态的、可交式的能力。从而基于静态的 HTML 页面将被取代，Web 页面可提供动态实时信息，并对客户操作进行反应。而且更重要的是，使用 JavaScript 语言可以在客户端实现一定量的动态的、交互式的操作，不需要通过网络将数据传送到服务器端进行计算后再通过网络反馈给客户端，从而既可大幅度降低服务器端的压力，又可以通过减少网络传输缩小系统反应延迟和网络流量。这些对于而今如何处理因特网上爆发性增长的海量数据是非常重要的。

JavaScript 语言是一种新的基于对象和事件驱动并具有安全性能的脚本语言，开发人

员可以在短时间内掌握,并应用到实际的软件开发中。JavaScript 语言在创建复杂程序和语言的易用性方面找到了一个最佳的平衡点。

JavaScript 语言不是解释型 Java 语言,二者虽然在名字上相近,但是由不同的公司发明,运行环境也大不相同。

Java 语言是由 Sun 公司开发的一种与平台无关的、面向对象的程序设计语言。Java 可以用来设计独立的应用程序,可以用来制作与平台无关的、与用户交互的复杂软件。Java 的源代码必须经过编译,使用 Java 语言开发的程序必须在 Java 虚拟机(JVM)内运行。

JavaScript 语言是由 Netscape 公司创建开发的。最初,Netscape 公司将其命名为 LiveScript,其语法以 Java 为基础,但比 Java 简洁,同时由于是一种脚本语言,无须编译,可直接由浏览器解释运行。后来见 LiveScript 大有发展前途,Sun 公司与 Netscape 公司一拍即合、签订协议,将 LiveScript 更名为 JavaScript,从而造就了这个强有力的 Web 开发语言。

后来,JavaScript 被提交给 ECMA(European Computer Manufacturers Association,欧洲计算机厂商协会)制定为 ECMAScript 标准,编号为 ECMA-262,目前最新的版本为 ECMA-262 3th Edition。ECMA-262 4th Edition 正在研讨开发过程中,Mozilla 基金会和 Adobe 公司是 ECMA-262 4th Edition 的主要推动者。

JavaScript 一般在一个浏览器内运行,是解释性编程语言,无需编译,而浏览器一般内置有 JavaScript 解释器,即解释引擎。

JavaScript 与 JScript 也是不一样的。JScript 是微软公司对 ECMA-262 语言规范的一种实现,即另一种脚本语言。而微软公司宣称 JScript 完全实现了 ECMA 标准。微软公司原本希望利用在操作系统上的优势使其发布的 VBScript 与 JavaScript 在客户端一决高下,但市场反应并不尽如人意;很快微软公司改变策略,根据 ECMA-262 规范发布了 JScript,并不断对其版本进行更新。JScript 最新的版本是基于尚未定稿的 ECMAScript4.0 版规范的 JScript .NET,并且可以在微软的.Net 环境下编译。

JavaScript 语言有以下几方面的特点:

- 脚本编写语言。首先 JavaScript 语言是一种脚本语言,同其他脚本语言一样,JavaScript 语言同样是一种解释性语言,它提供了一种简单而有效的开发方式。JavaScript 语言的基本结构形式与 C、C++、VB、Delphi 等高级语言类似,而 JavaScript 语言不需要实现编译。
- 基于对象的语言。JavaScript 语言是一种基于对象的语言。在编程时,JavaScript 语言可以运用自己已经创建的对象,许多功能可以来自于脚本环境中对象的方法与脚本的相互作用。
- 简单性。JavaScript 语言是一种基于 Java 基本语句和控制流之上的简单而紧凑的语言,另外其变量类型采用弱类型,并未使用严格的数据类型。JavaScript 语言入门很简单,但是要先学好 HTML 语言。
- 安全性。JavaScript 语言是一种安全性语言,它不允许访问本地的硬盘,并不能将数据存入到服务器上,不允许对网络文档进行修改和删除,只能通过浏览器实现信息

浏览或动态交互。从而可以有效地防止数据的丢失。

- 动态性。JavaScript 语言是动态的,它可以直接对用户或客户输入做出响应,无须经过服务器端程序。它对用户的反映响应,是以事件驱动的方式进行的。所谓事件驱动,就是指在主页中执行了某种操作所产生的动作,就称为“事件”。比如按下鼠标、移动窗口、选择菜单等都可以视为事件。当事件发生后,可能会引起相应的事件响应。
- 跨平台性。一般的程序设计语言编写的程序都与运行平台有关。例如,Linux 上无法运行 Windows 程序。JavaScript 语言是仅依赖于浏览器本身,与操作环境无太多关系,只要能运行浏览器的计算机,并支持 JavaScript 语言的浏览器就可以正确执行。任何事物都有其两面性,JavaScript 语言当然也有其局限性。
- 与浏览器相关的局限性。目前,因特网上有很多的浏览器,例如 Navigator、Internet Explore、Mosaic、Foxfire、Opera 等,虽然绝大多数浏览器都支持 JavaScript 语言,但并不是所有浏览器的所有版本都支持。当然在显示一个带有 JavaScript 程序的网页时,支持或不支持 JavaScript 语言的浏览器显示的结果将会全然不同。
- 与安全性相关的局限性。JavaScript 语言的设计目标之一就是“安全性”。它不允许访问本地的硬盘,并不能将数据存入到服务器上,不允许对网络文档进行修改和删除。总而言之,JavaScript 语言只生存于 WEB 主页的世界里。这自然要以牺牲 JavaScript 的一些功能来换得。

下面通过一个简单的例子进一步说明 JavaScript 语言。

这是利用记事本开发的一个 JavaScript 程序(例 4.1),显示如图 4.1 所示。

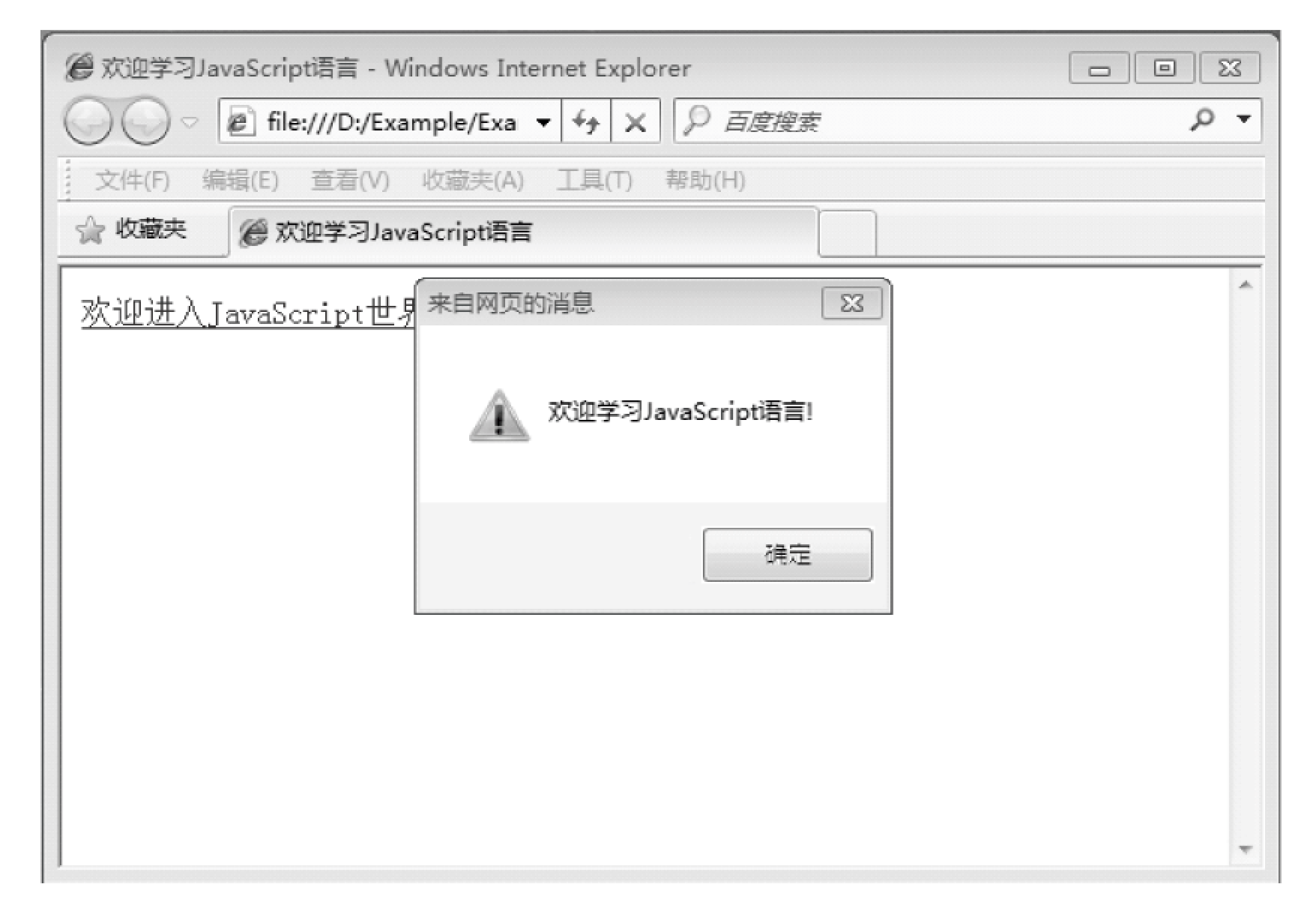


图 4.1 第一个 JavaScript 程序

建立该程序的方法是,选择开始→所有程序→附件→记事本,打开记事本,并输入程序如图 4.2 所示。

保存时,保存类型选择“所有文件”,文件名输入以 html 结尾的名称,如图 4.3 所示。

```
Example4-1.html - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
<html>
        <head>
                <title>欢迎学习JavaScript语言</title>
        <script language=JavaScript>
        function welcome() {
          alert("欢迎学习JavaScript语言!");}
        </script>
        </head>
        <body>
        <a href="#" onclick="welcome();">欢迎进入JavaScript世界！ </a>
        </body>
</html>
```

图 4.2　利用记事本编写 JavaScript 程序

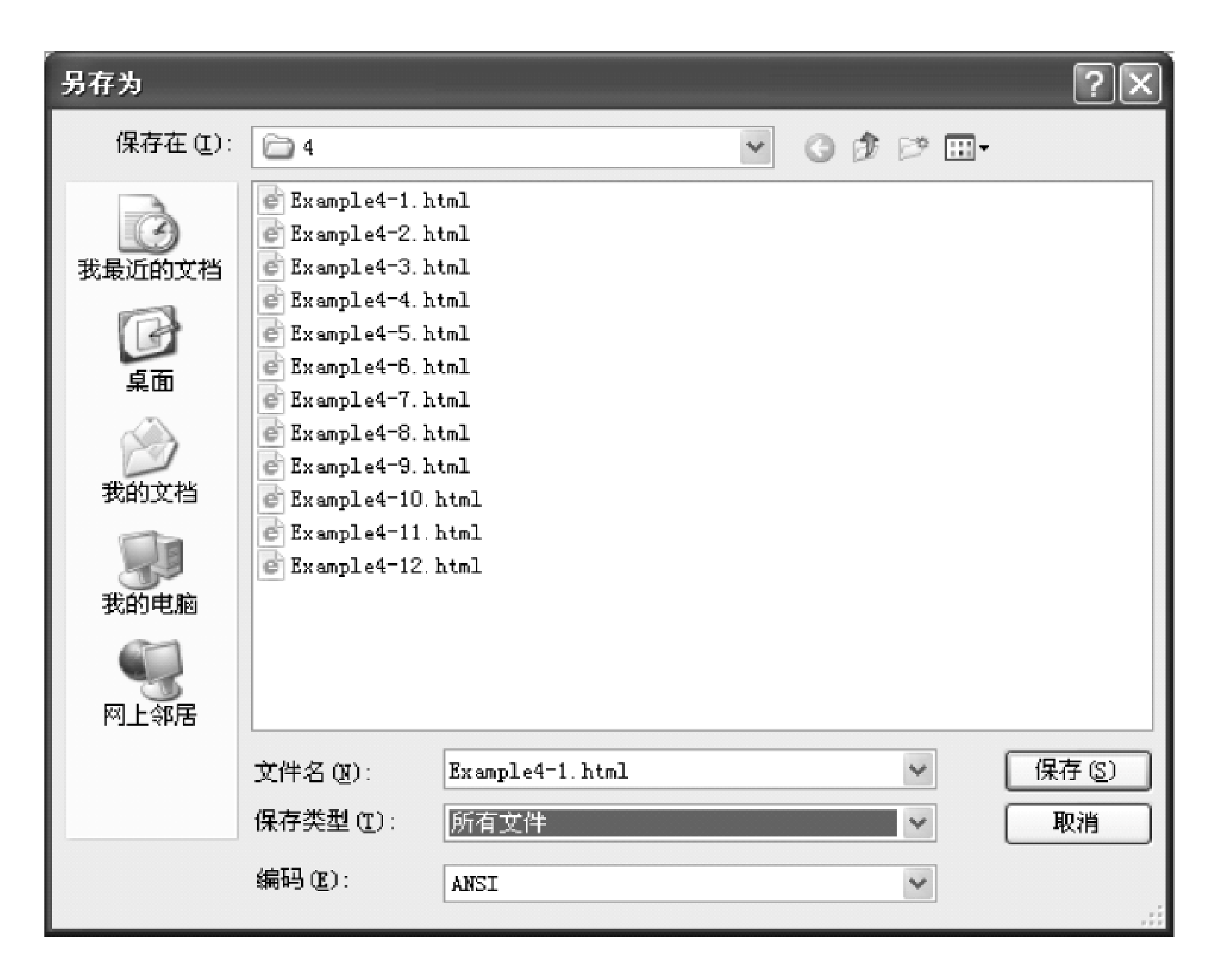


图 4.3　程序存储

其中在代码编辑的记事本窗口输入如下所示代码。

```
<html>
        <head>
                <title>欢迎学习 JavaScript 语言</title>
        <script language = JavaScript>
        function welcome() {
          alert("欢迎学习 JavaScript 语言!");}
        </script>
```

```
        </head>
        <body>
        <a href = "#" onclick = "welcome();">欢迎进入 JavaScript 世界!</a>
        </body>
</html>
```

<script language = JavaScript>代表代码的开始,</script>代表代码的结束,JavaScript 程序代码要放在两者之间。"alert("欢迎学习 JavaScript 语言!");"是真正的 JavaScript 程序,alert 将弹出一个对话框,"欢迎学习 JavaScript 语言!"将是对话框中要显示的内容。

JavaScript 也可以存放在外部脚本文件.js 中。例如 JavaScript 语句"alert("欢迎学习 JavaScript 语言!");"可存放在外部脚本文件 Example4-1.js 中。如果设计页面需用 Example4-1.js 时,可以下列语句调用。

```
<script language = "JavaScript" src = "Example4 - 1.js"></script>
```

同其他语言一样,JavaScript 语言有其自身的基本数据类型、表达式和算术运算符以及程序的基本框架结构。

变量是程序设计语言的最基本元素。在程序运行时,需要用一个个单元将信息存储起来,这些单元就是变量。变量可以理解为一个存放信息或数据的容器。JavaScript 语言是一种弱类型语言,即程序中所用到的每个存储单元(即变量)不必指明其数据类型。

JavaScript 语言对于变量的命名有一些规定。首先,在 JavaScript 语言中定义的变量名称不能与保留字冲突。保留字是指在 JavaScript 语言中已经使用的名称以及保留给以后的版本可能使用的单词。例如,do,function 等。其次,变量名必须以字母或者下划线开始,而且变量名中不能有空格。另外,JavaScript 语言是区分大小写的。

常量是 JavaScript 中的固定值,其在程序中是不发生变化的,为程序提供固定的和精确的值。变量是在程序中可以赋值并改变的量,变量的值可以在程序运行时发生变化。JavaScript 中变量的生存期有两种:全局变量和局部变量。全局变量在主程序中定义,其有效范围从其定义开始,一直到本程序结束为止。局部变量在程序的函数中定义,其有效范围仅在该函数之中;当函数结束后,局部变量生存期也随即结束。

4.1.1 基本数据类型

JavaScript 语言中有 4 种基本数据类型:数值类型、逻辑类型、字符串类型、未定义类型。

数值类型。数值类型是 JavaScript 语言中比较常用的类型。在 JavaScript 语言中数值类型既可表示整数,也可以表示实数。这与其他编程语言可能不同。

逻辑类型。逻辑类型主要用于逻辑运算,因此逻辑类型的变量只有两个值:True 或 False。

字符串类型。一个字符串类型常量是由单引号或者双引号包围的多个字符组成的,注意这里的引号指的是英文引号。

未定义类型。未定义类型又称为 undefined 类型,表示一个变量还没有赋予任何初值。

4.1.2 运算符

运算符可以用来处理数字、字符串及比较运算等,是组成表达式的连接黏和剂。与大多数编程语言一样,JavaScript 语言的运算符可以被分成几种常见类型。

1. 算术运算符

算术运算符主要用于数据计算,其操作数一般为数值类型,返回结果也是一个数值类型。在 JavaScript 语言中算术操作符又可以分为单目操作运算符和双目操作运算符;运算所需变量为一个的运算符叫单目操作运算符,运算所需变量为两个的运算符叫双目操作运算符。JavaScript 语言算数运算符如表 4.1 所示。

表 4.1 算术运算符

操作符	类别	示　例	说　明
＋	二元	m1＝m2＋m3	m1 等于 m2 与 m3 的和
－	二元	m1＝m2－m3	m1 等于 m2 减去 m3 的差
*	二元	m1＝m2 * m3	m1 等于 m2 与 m3 的乘积
/	二元	m1＝m2/m3	m1 等于 m1 除以 m2 的值
%	二元	m1＝m2%m3	m1 等于 m2 除以 m3 的余数
＋＋	一元	分＋＋m1 与 m1＋＋	前者先对 m1 加 1 而后操作,后者先操作后加 1
－－	一元	分－－m1 与 m1－－	前者先对 m1 减 1 而后操作,后者先操作后减 1

2. 赋值运算符

赋值运算符是用于将一个数据赋予一个变量、属性或引用,其中数据可以是变量,也可以是表达式。常用的赋值运算符如表 4.2 所示。

表 4.2 赋值运算符

操作符	类别	示　例	说　明
＝	二元	m1＝m2	m1 被赋予 m2 的值
＋＝	二元	m1＋＝m2	m1 被赋予 m2 与 m1 的和
－＝	二元	m1－＝m2	m1 被赋予 m2 与 m1 的差
*＝	二元	m1 *＝m2	m1 被赋予 m2 与 m1 的积
/＝	二元	m1/＝m2	m1 被赋予 m1 除以 m2 的值
%＝	二元	m1%＝m2	m1 被赋予 m2 除以 m2 的余数

3. 关系运算符

关系运算符也就是比较运算符,返回值是布尔型的真(true)或假(false),如表 4.3 所示。

表 4.3 关系运算符

操作符	类别	示 例	说 明
＝＝	二元	m1＝m2＝＝m3	若 m2 和 m3 相等，则 m1 为 true，否则为 false
！＝	二元	m1＝m2！＝m3	若 m2 和 m3 不相等，则 m1 为 true，否则为 false
＞	二元	m1＝m2＞m3	若 m2 大于 m3，则 m1 为 true，否则为 false
＜	二元	m1＝m2＜m3	若 m2 小于 m3，则 m1 为 true，否则为 false
＞＝	二元	m1＝m2＞＝m3	若 m2 不小于 m3，则 m1 为 true，否则为 false
＜＝	二元	m1＝m2＜＝m3	若 m2 不大于 m3，则 m1 为 true，否则为 false

4. 逻辑运算符

逻辑运算符主要用于逻辑判断，主要包括逻辑与、逻辑或和逻辑非，如表 4.4 所示。

表 4.4 逻辑运算符

操作符	类别	示 例	说 明
&&	二元	m1＝m2&&m3	若 m2 和 m3 都为 true，则 m1 为 true，否则为 false
｜｜	二元	m1＝m2｜｜m3	若 m2 或 m3 有一个为 true，则 m1 为 true，否则为 false
!	一元	!m1	若 m1 为 true，则!m1 为 false，否则为 true

5. 位运算符

位运算符用于对操作数进行位运算。在计算机中，信息是以二进制形式存储的。JavaScript 中位操作符就是专门针对二进制数据运算的。

- 按位与(&)：按二进制进行与操作。
- 按位或(|)：按二进制进行或操作。
- 按位异或(^)：按二进制进行异或操作。
- 按位取反(～)：按二进制进行取反操作。
- 左移(＜＜)：按二进制进行左移操作。
- 右移(＞＞)：按二进制进行与操作。
- 添 0 右移(＞＞＞)：添 0 后右移。

4.1.3 基本控制语句

实际生活中，并非所有的事情都是按部就班地进行。基本的数据类型和流程控制语句是任何变成语言都必须有的内容。与其他编程语言类似，在 JavaScript 中可通过控制语句控制程序的流程，从而形成程序的分支和循环。

1. 选择语句

选择语句即分支语句，主要用于根据某个条件来决定哪个流程被执行。

(1) if 语句

if 语句的格式如下所示。

```
if(布尔型表达式)
{
   语句 1;
}
else
{
   语句 2;
}
```

当布尔型表达式的值为 true 时，执行语句 1。否则跳过语句 1，若有 else 语句，则执行语句 2。

（2）switch 语句

switch 语句非常类似于 if 语句，switch 语句可以根据表达式的值有条件地执行分支流程。switch 语句可以一次将表达式的值与多个值进行比较，这是 switch 语句与 if 语句不同的地方。switch 语句的格式如下所示。

```
switch(表达式 1)
{
   case 值 1:
            语句 1;
            break;
case 值 2:
            语句 2;
            break;
   …
   default:
            默认语句;
            break;
}
```

switch 语句将表达式 1 的值分别与 case 中的多个值进行比较。若有相符的选项，则执行相应的语句集；否则就执行默认语句集。

例 4.1 说明 if 语句与 switch 语句的用法，程序如下所示。

```
<html>
        <head>
                <title>if 语句与 switch 语句</title>
        </head>
        <body>
        <script language = JavaScript>
        var d = "Saturday";
        document.write("d = " + d + "<br>");
        if(d == "Saturday")
          document.write("d 是周末");
        else if(d == "Sunday")
          document.write("d 是周末");
        else
          document.write("d 不是周末");
        document.write("<br>");
```

```
        switch(d)
        {case "Saturday":document.write("d是周末");
                         break;
         case "Sunday":document.write("d是周末");
                         break;
         default:document.write("d不是周末");
         }
        </script>
        </body>
</html>
```

运行效果如图4.4所示。

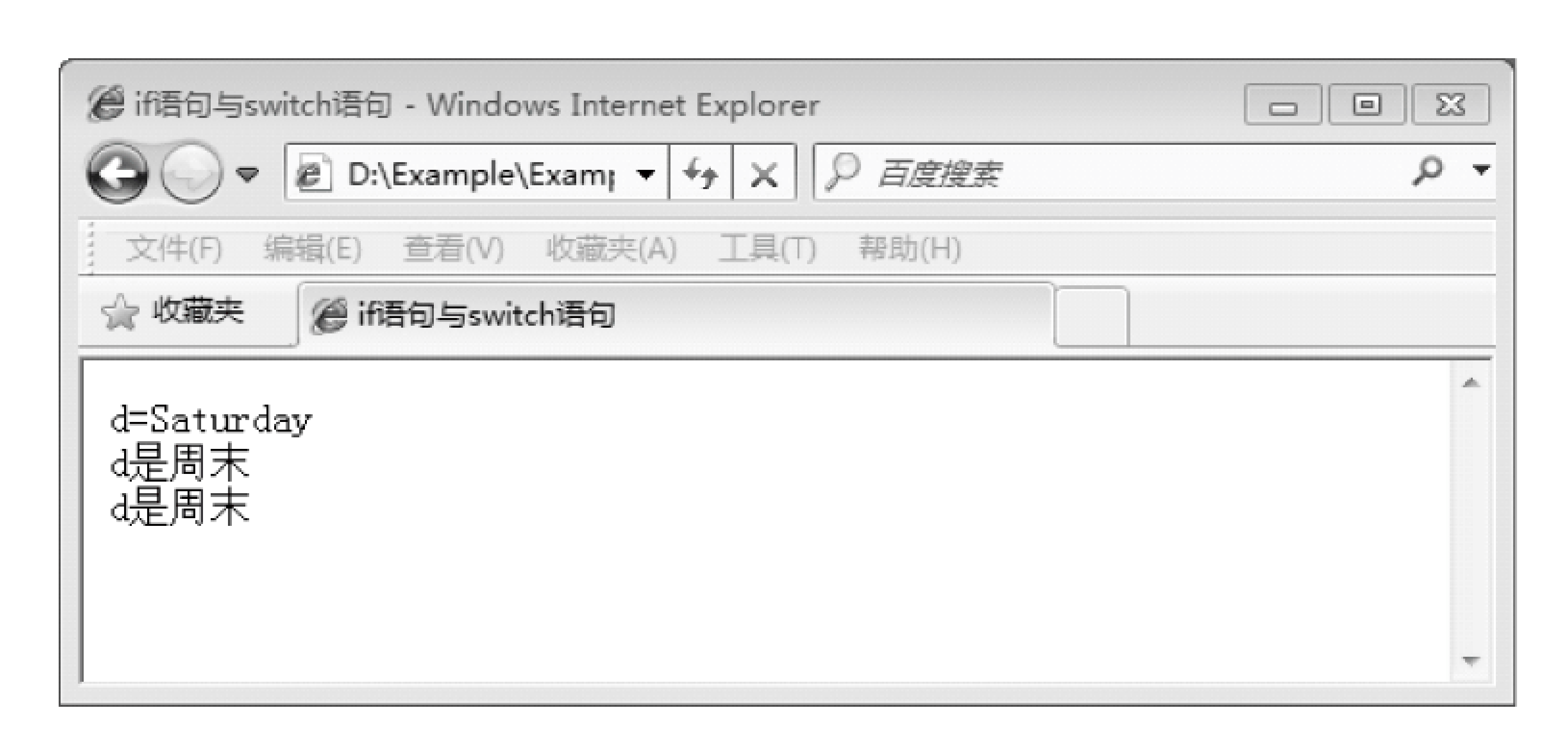


图4.4 if语句与switch语句

提示：switch语句的各个case子语句中，一般都要以“break;”语句结束。

2. 循环语句

当需要进行大量重复操作时，通过循环语句可以创建循环。JavaScript语言中提供了以下循环控制语句：while语句、do-while语句、for语句等。

（1）while语句

while语句执行一个语句或语句块，直到条件表达式为false为止。while语句的格式如下所示。

```
while(条件表达式)
{
    循环代码;
}
```

break等语句可以将控制传递到循环之外，从而终止while循环。若需要将控制传递给下一个循环但不退出循环，可以使用continue语句。

（2）do-while语句

do-while语句会根据条件表达式的值有条件地执行其中的循环代码。do-while循环非常类似于while循环。

do-while循环与while循环的重要区别在于：while循环的布尔测试是在循环刚开始时进行的，而do-while循环的布尔测试是在最后进行的，这样do-while循环中的循环语句将

至少会执行一次。

do…while 语句的格式如下所示。

```
do
{
    循环代码;
} while(条件表达式)
```

(3) for 循环

for 循环重复执行一个语句或语句块，直到指定的循环变量超过循环条件为止。for 语句是最常使用的循环语句。

for 语句的格式如下所示。

```
for(循环变量初始化; 循环条件; 循环操作)
{
    循环代码;
}
```

for 语句的执行过程如下所示。

① 运行初始化语句，为循环变量赋一个初值；

② 判断循环条件，决定是否进入循环；

③ 若进入循环，执行循环语句。然后转②。

(4) break 和 continue 语句

break 语句使得循环从 For 或 while 中跳出，continue 使得跳过循环内剩余的语句而进入下一次循环。

例 4.2 说明循环语句的用法，程序如下所示。

```
<html>
        <head>
                <title>循环语句</title>
        </head>
        <body>
        <script language = JavaScript>
        var x,y;
        var s;
        document.write("<h1>九九乘法表</h1>");
        for(x = 1;x <= 9;x++)
        { for(y = 1;y <= x;y++)
          { s = x + " * " + y + " = " + (x * y);
          document.write(s + "  "); }
          document.write("<br>");
        }
        </script>
        </body>
</html>
```

运行效果如图 4.5 所示。

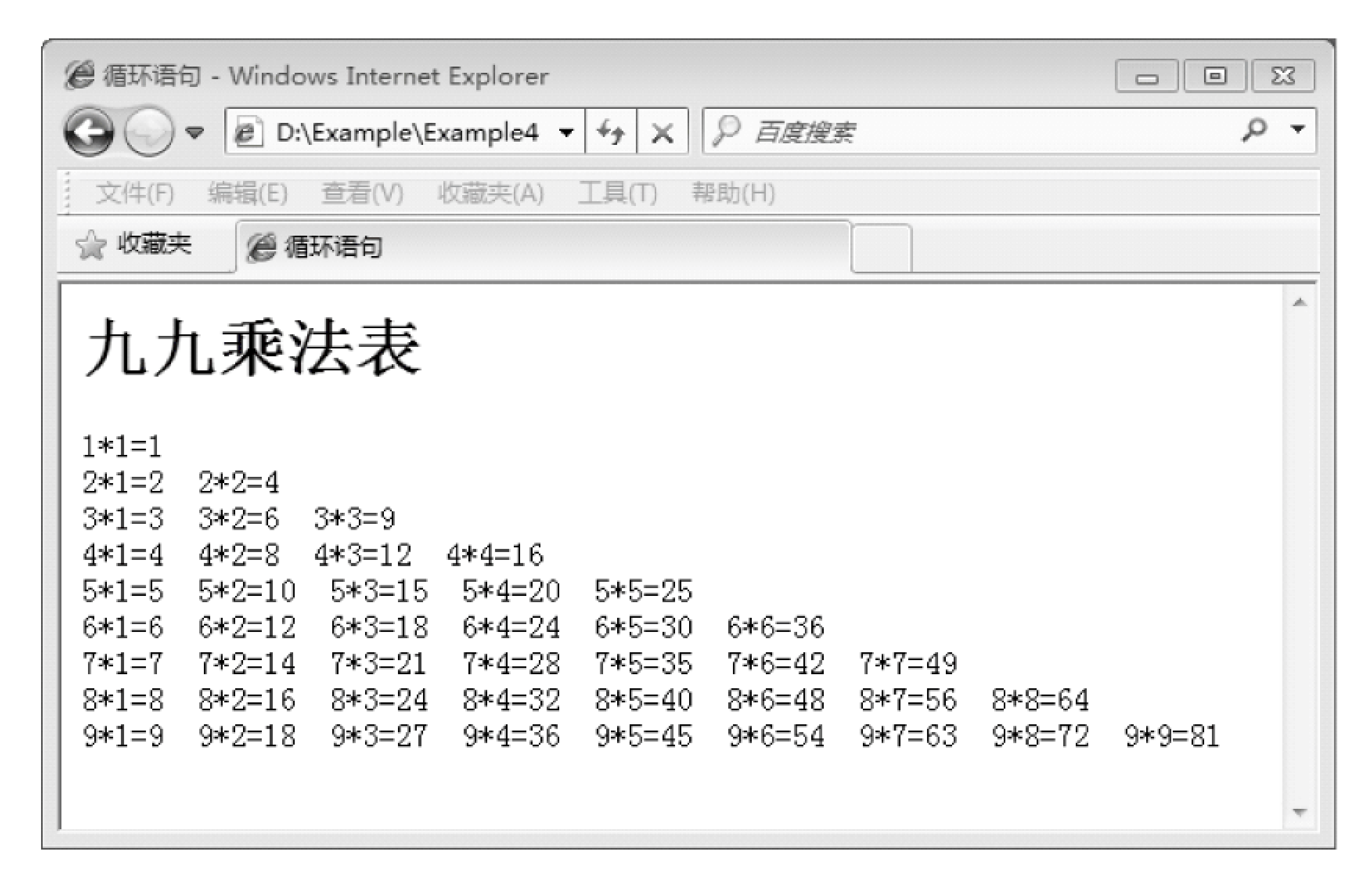


图 4.5 循环语句

提示：该例使用双重循环，外层循环控制行数；内层循环控制每行的内容，包括被乘数、乘号、乘数、等号、积以及换行。

4.1.4 函数

函数是能够完成一定功能的代码块组合。在 JavaScript 中，函数可以被事件或其他语句调用。在程序设计时，经常把一个大程序分为若干个小程序，即模块，每个模块完成一定的功能且相对独立。从而使各模块相互独立，任务单一，程序清晰，易懂、易读、易维护。而且这样设计可以解决代码的重复。可以把经常用到的完成某种功能的程序段编写成函数，每当需要实现该功能时只要调用该函数即可，无须重复编写代码。

在 JavaScript 中，不区分函数(Function)和过程(Procedure)，只有函数。函数完成相应功能后可以有返回值，也可以没有，即 JavaScript 中的函数具有函数和过程的功能。给函数传递信息的方法是通过函数名后的参数来传递的，即要传递给函数的信息就是参数。

提示：当然，函数可以使用参数，也可以不使用参数。函数参数的个数没有限制，由该函数的具体功能而决定。

1. 函数定义

在 JavaScript 中，函数必须在定义以后才能使用。通常函数的定义放在 HTML 文档头中。函数定义的格式如下所示。

```
function 函数名 (参数,变元){
函数体;
return 表达式;
}
```

提示：函数名是定义自己函数的名字；参数是传递给函数使用或操作的值；返回值前用关键字 return。

例 4.3 说明函数的定义与使用，程序如下所示。

```
<html>
        <head>
                <title>函数的定义与使用</title>
        <script language = JavaScript>
        //函数定义
        function test()                         //无参数函数
        {
          document.write("该函数没有使用参数");
         }
        function test1(exp1)                    //带参数函数
        {
          document.write("该函数使用参数" + exp1);
         }
        </script>
        </head>
        <body>
        <script language = JavaScript>
        //函数使用
        test();
        document.write("<br>");
        test1("x");
        </script>
        </body>
</html>
```

运行效果如图 4.6 所示。

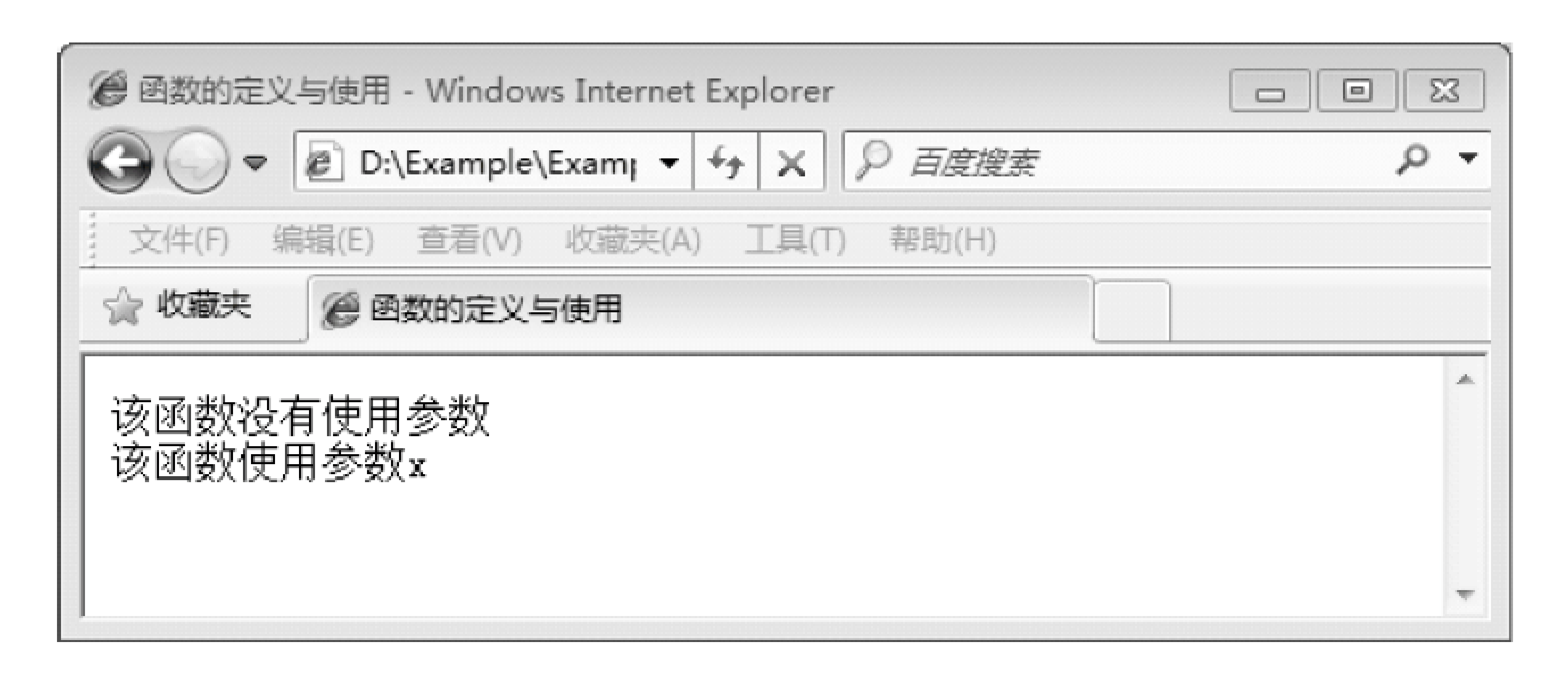


图 4.6 函数的定义与使用

2. 函数中的形式参数

在函数的定义中，函数名后有参数表，参数个数可能是一个或几个，在 JavaScript 中可以通过 arguments. length 来确定参数个数。

3. JavaScript 的函数

在 JavaScript 中支持两种函数：JavaScript 中的库函数和用户自己创建的函数。

(1) 库函数。JavaScript 中提供了很多库函数，以供编程人员直接使用。例如，trim()

正则表达式去除左右空格；isDigit()检查是否为数字；eval()对用字符串行表示的合法表达式求值。

(2) 用户自己创建的函数。在JavaScript中允许用户自己创建自己的函数，并在需要的地方使用，函数的创建和使用可参考例4.4。

4.1.5 事件驱动和事件处理

JavaScript语言是一种基于对象的语言，其基本特征就是采用事件驱动。用户的操作或系统操作的动作称为事件(Event)。由事件引发的一连串程序的动作，称为事件驱动(Event Driver)。而对事件进行处理的程序或函数称为事件处理程序(Event Handler)。

1. 事件处理程序

在JavaScript语言，事件处理程序通常由函数担任，其基本格式与函数格式一样，如下所示。

```
function 事件处理名(参数表){
事件处理语句集;
}
```

2. 事件驱动

在JavaScript语言中，事件主要是通过鼠标或键盘的动作引发的，其主要事件如表4.5所示。

表4.5 主要事件

事 件	动 作	事 件	动 作
onAbort	中止正在加载的对象	onUnload	关闭当前网页
onBlur	失去焦点	onMouseDown	按下鼠标左键
onFocus	获取焦点	onMouseMove	移动鼠标指针
onChange	改变对象的值	onMouseOut	鼠标指针离开某对象
onClick	在对象上单击鼠标	onMouseOver	鼠标指针悬停于某对象之上
onDblClick	在对象上双击鼠标	onMouseUp	放开鼠标左键
onDrogDrop	拖拽对象	onMove	窗口被移动时
onError	加载文件或图形发生错误	onResize	窗口大小被改变
onKeyDown	按下键盘上任意键的瞬间	onSelect	选择某对象
onKeyPress	按下键盘上的任意键时	onSubmit	单击表单上的Submit按钮
onKeyUp	某键被按下后弹起瞬间	onReset	单击表单上的Reset按钮
onLoad	浏览器读入文件时		

3. 事件处理程序举例

(1) 自动加载和自动卸载

下例是一个自动加载和自动卸载的例子。当浏览器读入网页时将自动加载loadform()函数，关闭网页时自动运行卸载unloadform()函数，例4.5的程序如下所示。

```
<html>
        <head>
                <title>自动加载与自动卸载</title>
        <script language = JavaScript>
        function loadform()                         //自动加载函数
        {
          alert("这是一个自动加载的例子!");
         }
        function unloadform()                       //自动卸载函数
        {
          alert("这是一个自动卸载的例子!");
         }
        </script>
        </head>
        <body onLoad = "loadform()" onUnload = "unloadform()">
        <h1>自动加载与自动卸载例子<h1>
        </body>
</html>
```

运行效果如图 4.7 所示，当浏览器运行网页时将弹出一个对话框“这是一个自动加载的例子”。

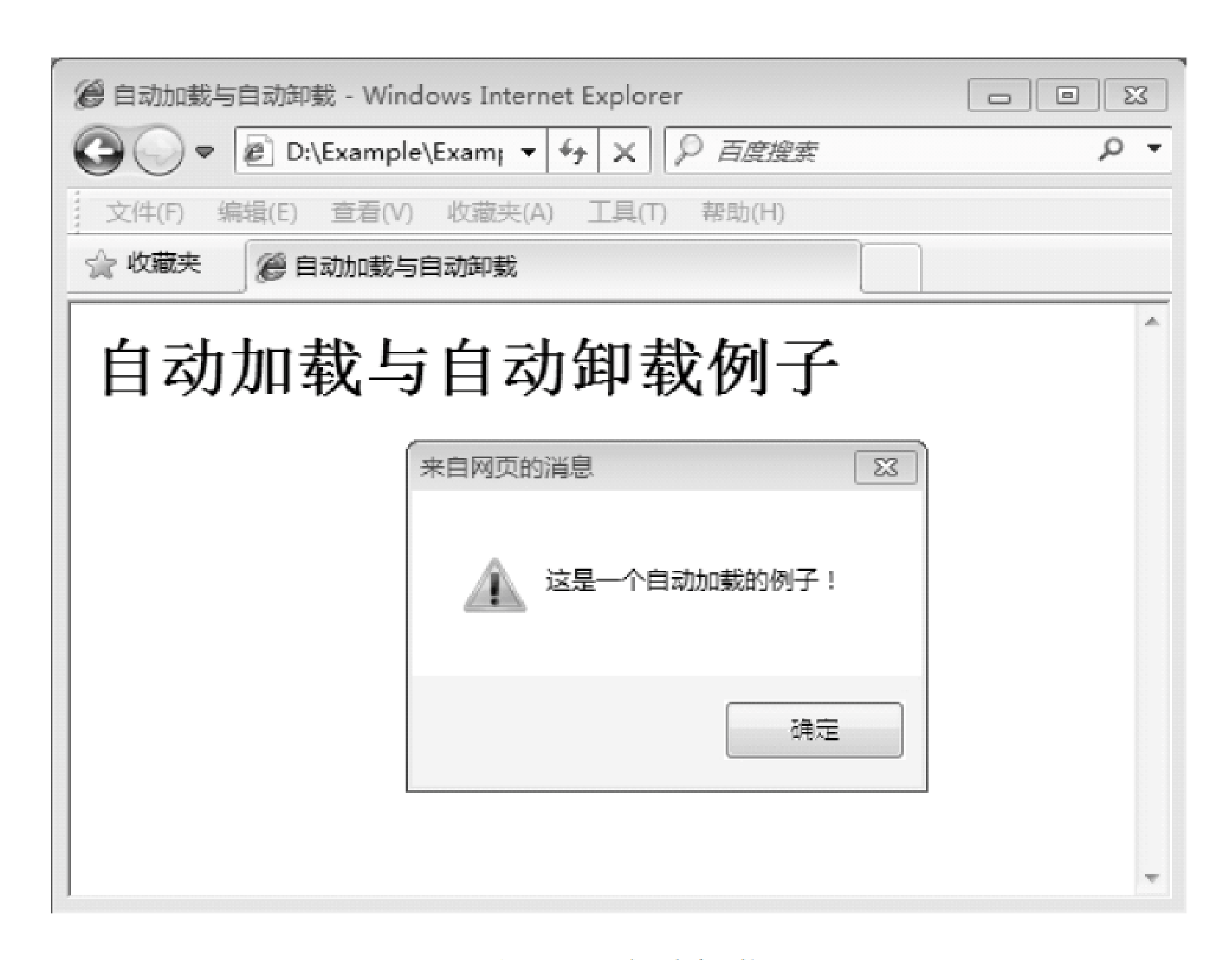


图 4.7　自动加载

另外，当关闭本页面时会将弹出一个对话框“这是一个自动卸载的例子”，运行效果如图 4.8 所示。



图 4.8　自动卸载

(2) 事件驱动和函数处理例子

下例是一个事件驱动和函数处理的例子。当用户输入姓名和单位后，单击“新增”按钮，在下面表格中将根据用户输入信息添加新的一行，例 4.6 的程序如下所示。

```
<html>
```

```
<head>
<title>事件驱动和函数处理的例子</title>
<script language="JavaScript"><!--
  function addRow() {
    var tableObj = document.getElementById("mainTb");
    var tableBodyObj = document.getElementById("mainBody");
    var newRowObj = document.createElement("tr");
    newRowObj.id = "row" + (tableObj.rows.length-1);
    var newNameCell = document.createElement("td");
    var newCompanyCell = document.createElement("td");
    var newButtonCell = document.createElement("td");
    newNameCell.innerHTML = document.getElementById("newName").value;
    newCompanyCell.innerHTML = document.getElementById("newCompany").value;
    newButtonCell.innerHTML = '<input type="button" value="删除" onclick="deleteRow
('+(tableObj.rows.length-1)+')">';

    newRowObj.appendChild(newNameCell);
    newRowObj.appendChild(newCompanyCell);
    newRowObj.appendChild(newButtonCell);
    tableBodyObj.appendChild(newRowObj);
  }
  function deleteRow(index) {
    var tableBodyObj = document.getElementById("mainBody");
    var rowObj = document.getElementById('row'+index);
    tableBodyObj.removeChild(rowObj);
  }
--></script>
</head>
<body style="margin:40px">
<div><span id="new">姓名<input type="text" name="newName" id="newName">单位<input
type="text" id="newCompany"><input type="button" value="新增" onclick="addRow()">
</span></div>
<table id="mainTb" border="1">
  <tbody id="mainBody">
    <tr><th width="80">姓名</th><th width="250">单位</th><th>操作</th></tr>
    <tr id="row0"><td>张三</td><td>上海软件公司</td><td><input type="button" value
="删除" onclick="deleteRow(0)"></td></tr>
    <tr id="row1"><td>李四</td><td>广州食品公司</td><td><input type="button" value
="删除" onclick="deleteRow(1)"></td></tr>
  </tbody>
</table>
</body>
</html>
```

运行效果如图4.9所示，用户可以在页面中输入姓名与单位，当单击“新增”时在下面的表格中将会增加相应的内容；当用户单击表格中某一行的“删除”时，相应的行就会从表格中删除。



图 4.9 事件驱动和函数处理

4.1.6 基于对象的 JavaScript 语言

JavaScript 语言是一种基于对象的语言。JavaScript 提供了一些非常有用的预定义对象以供编程人员使用，但 JavaScript 没有提供抽象、继承、重载等有关面向对象语言的功能。

1. 对象的基础知识

JavaScript 语言中的对象是由属性（Properties）和方法（Methods）两个基本的元素的构成的。属性是对象在实施其所需要行为的过程中，实现信息的装载单位，从而与变量相关联；方法是指对象能够按照设计者的意图而被执行，从而与特定的函数相联。

JavaScript 语言中的对象可以通过以下方式获得：直接引用 JavaScript 内部对象；由浏览器环境中提供；创建新对象。

提示：一个对象在被引用前，这个对象必须存在，否则将出错。

2. 对象操作语句

（1）new 运算符

编程人员可以使用 new 运算符来创建一个对象，其创建对象使用如下格式：

```
newobject = new Object(参数表);
```

下例是一个关于创建新对象和使用对象的例子，例 4.7 的程序如下所示。

```
<html>
<head>
<title>创建新对象和使用对象</title>
</head>
<body>
<script language = "JavaScript">
var today;
today = new Date();
```

```
document.write("今天是" + today.getFullYear() + "年" + (today.getMonth() + 1) + "月" + today.getDate() + "日");
</script>
</body>
</html>
```

其中 today 是利用 Date()新创建的对象,可以利用 today.getFullYear、today.getMonth、today.getDate 得到其年、月、日。运行效果如图 4.10 所示,当浏览器读入网页时将弹出一个对话框“这是一个自动加载的例子”。



图 4.10 创建新对象和使用对象

(2) with 语句

使用 with 语句的意思是:在该语句体内,任何对变量的引用被认为是这个对象的属性,以节省一些代码。格式如下:

```
with object
{ 语句集; }
```

所有在 with 语句后的花括号中的语句,都是在后面 object 对象的作用域的。

例如如下语句:

```
<script language = "JavaScript">
      document.write ("欢迎光临:<br>");
      document.write ("一楼是家电卖场.<br>");
      document.write ("二楼是计算机卖场.<br>");
</Script>
```

可以使用 with 语句来改写成如下代码:

```
<script language = "JavaScript">
   with (document)
     {
      write ("欢迎光临:<br>");
      write ("一楼是家电卖场.<br>");
      write ("二楼是计算机卖场.<br>");
     }
</Script>
```

(3) this 关键字

this 是对当前的引用,在 JavaScript 由于对象的引用是多层次、多方位的,往往一个对

象的引用又需要对另一个对象的引用,而另一个对象有可能又要引用另一个对象,这样有可能造成混乱,为此 JavaScript 提供了一个用于将对象指定当前对象的语句 this。随着函数使用场合的不同,this 的值会发生变化。但是有一个总的原则,那就是 this 指的是调用函数的那个对象。

(4) for…in 语句

格式如下:

```
for(对象属性名 in 已知对象名)
```

该语句的功能是用于对已知对象的所有属性进行操作的控制循环。它是将一个已知对象的所有属性反复赋值给一个变量,而不是使用计数器来实现的。

下例是一个关于 for…in 语句的例子,例 4.8 的程序如下所示。

```
<html>
        <head>
           <title>for…in 语句</title>
        </head>
        <body>
        <script language = JavaScript>
        var x;
        var mycars = new Array();
        mycars[0] = "新浪网";
        mycars[1] = "百度网";
        mycars[2] = "搜狐网";
        for(x in mycars)
        {
        document.write(mycars[x] + "<br>");
         }
        </script>
        </body>
</html>
```

运行效果如图 4.11 所示。

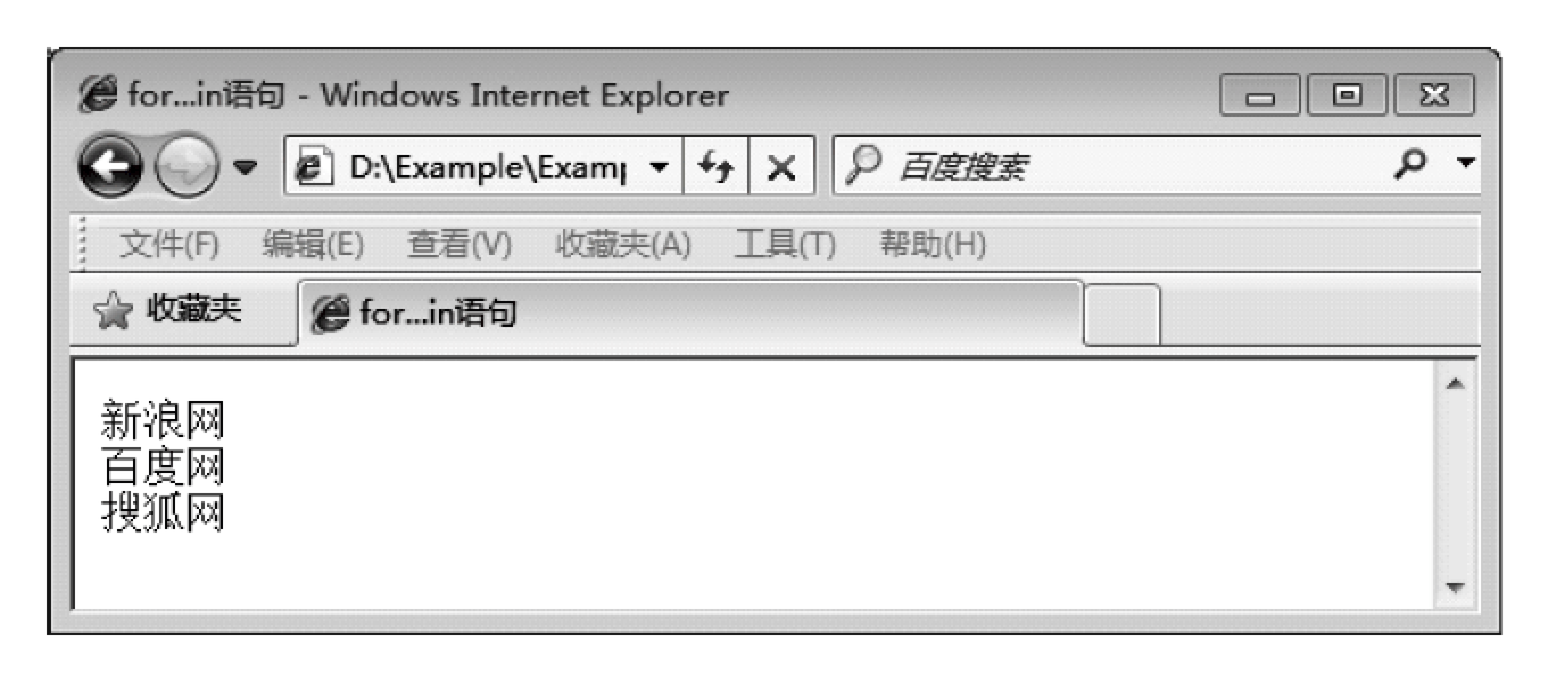


图 4.11 for…in 语句

3. 常用对象的属性与方法

JavaScript 为编程人员提供了一些非常有用的常用内部对象和方法,例如 String(字符

串)、Math(数值计算)、Date(日期)等。

(1) String(字符串)

string 对象的属性主要是 length,它表明了字符串中的字符个数,包括所有符号。

string 对象的方法主要用于有关字符串在 Web 页面中的显示、字体大小、字体颜色、字符的搜索以及字符的大小写转换。包括以下方法:

anchor() 创建 HTML 锚。

big() 用大号字体显示字符串。

blink() 显示闪动字符串。

bold() 使用粗体显示字符串。

charAt() 返回在指定位置的字符。

charCodeAt() 返回在指定的位置的字符的 Unicode 编码。

concat() 连接字符串。

fixed() 以打字机文本显示字符串。

fontcolor() 使用指定的颜色来显示字符串。

fontsize() 使用指定的尺寸来显示字符串。

fromCharCode() 从字符编码创建一个字符串。

indexOf() 检索字符串。

italics() 使用斜体显示字符串。

lastIndexOf() 从后向前搜索字符串。

link() 将字符串显示为链接。

localeCompare() 用本地特定的顺序来比较两个字符串。

match() 找到一个或多个正在表达式的匹配。

replace() 替换与正则表达式匹配的子串。

search() 检索与正则表达式相匹配的值。

slice() 提取字符串的片断,并在新的字符串中返回被提取的部分。

small() 使用小字号来显示字符串。

split() 把字符串分割为字符串数组。

strike() 使用删除线来显示字符串。

sub() 把字符串显示为下标。

substr() 从起始索引号提取字符串中指定数目的字符。

substring() 提取字符串中两个指定的索引号之间的字符。

sup() 把字符串显示为上标。

toLocaleLowerCase() 把字符串转换为小写。

toLocaleUpperCase() 把字符串转换为大写。

toLowerCase() 把字符串转换为小写。

toUpperCase() 把字符串转换为大写。

toSource() 代表对象的源代码。

toString() 返回字符串。

valueOf() 返回某个字符串对象的原始值。

下面是一个打字机效果的例子，显示的文字如打字机打字一样一个一个地显示出来。例 4.9 的程序如下所示。

```
<html>
        <head>
                <title>打字机效果</title>
        <script language = JavaScript>
        var j = 0;
        function textlist() {
        j = textlist.arguments.length;
        for (i = 0; i<j; i++)
        this[i] = textlist.arguments[i];
        }
        tl = new textlist("JavaScript 语言是由 Netscape 公司开发的","最初,Netscape 公司将其命名为 LiveScript","见到 LiveScript 大有发展前途,Sun 公司与 Netscape 公司一拍即合、签订协议,将 LiveScript 更名为 JavaScript","从而造就了这个强有力的 Web 开发语言.");
        var x = 0;
        pos = 0;
        var l = tl[0].length;
        function tp()
        { document.typeform.typefield.value = tl[x].substring(0, pos) + "_";
          if(pos++ == l)
          {  pos = 0;
             setTimeout("tp()", 1000);
             if(++x == j) x = 0;
             l = tl[x].length;
          }
          else
          { setTimeout("tp()", 50);}
        }
        </script>
        </head>
       <body OnLoad = "tp()">
         <form name = typeform>
           <textarea name = typefield rows = 10 cols = 50 >这里显示打字机效果</textarea>
</form>
       </body>
</html>
```

运行效果如图 4.12 所示。

(2) Math(数值计算)

如果要进行数学运算可以使用前面讲过的运算符，但它们只能用来进行简单的加减乘除等运算；如果要进行复杂的数学运算，可以使用 Math 对象。Math 对象并不像 Date 和 String 那样是对象的类，因此没有构造函数 Math()，像 Math. sin()这样的函数只是函数，不是某个对象的方法。无须创建，通过把 Math 作为对象使用就可以调用其所有属性和方法。

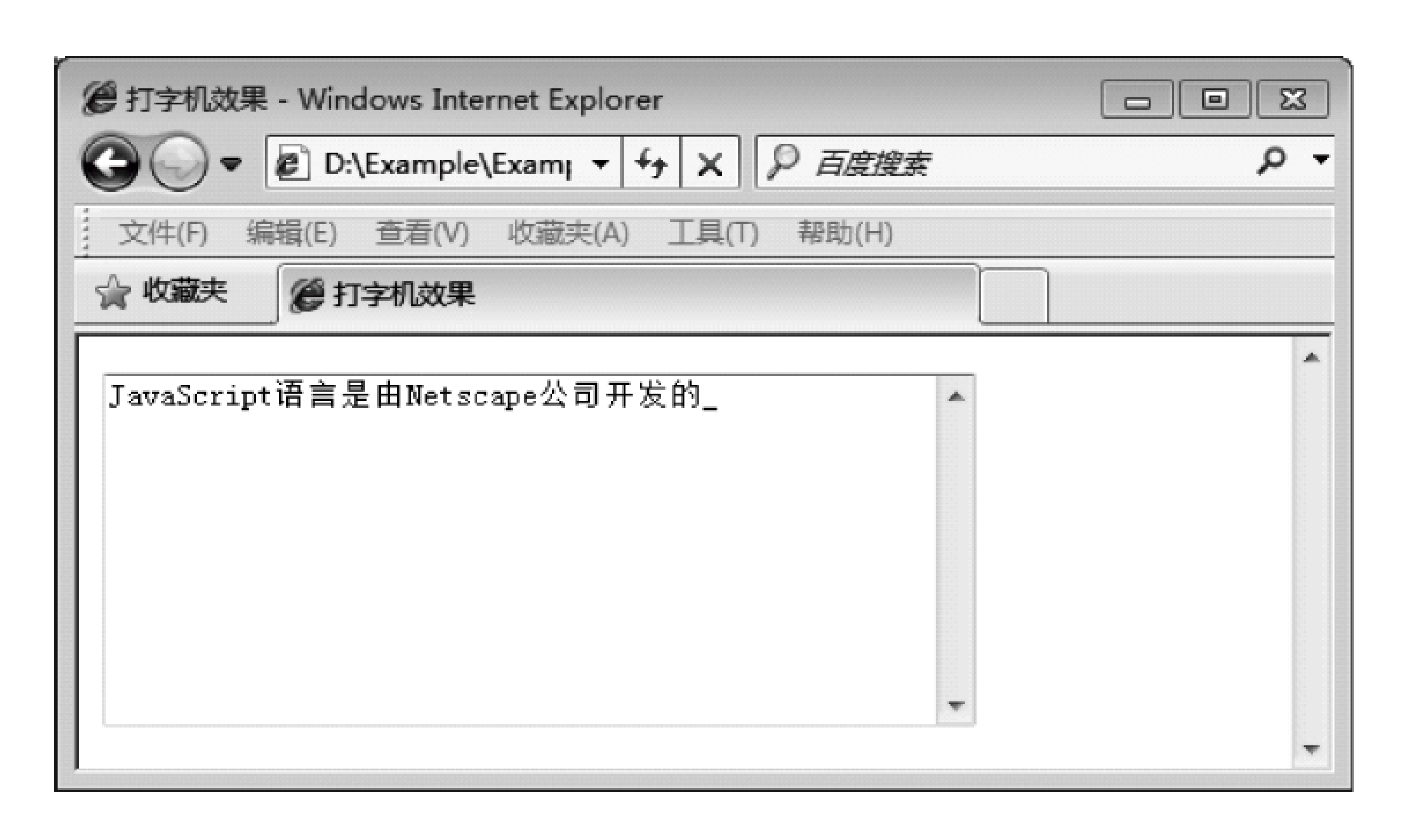


图 4.12　打字机效果

Math 的属性有：

E：欧拉常量，自然对数的底(约等于 2.718)。

LN2：2 的自然对数(约等于 0.693)。

LN10：10 的自然对数(约等于 2.302)。

LOG2E：以 2 为底的 e 的对数(约等于 1.442)。

LOG10E：以 10 为底的 e 的对数(约等于 o.434)。

PI：π 的值(约等于 3.14159)。

SQRT1_2：0.5 的平方根(约等于 o.707)。

SQRT2：2 的平方根(约等于 1.414)。

Math 的方法有：

abs(x)：返回数的绝对值。

acos(x)：返回数的反余弦值。

asin(x)：返回数的反正弦值。

atan(x)：以介于－PI/2 与 PI/2 弧度之间的数值来返回 x 的反正切值。

atan2(y,x)：返回从 x 轴到点(x,y)的角度(介于－PI/2 与 PI/2 弧度之间)。

ceil(x)：对数进行上舍入。

cos(x)：返回数的余弦。

exp(x)：返回 e 的指数。

floor(x)：对数进行下舍入。

log(x)：返回数的自然对数(底为 e)。

max(x,y)：返回 x 和 y 中的最高值。

min(x,y)：返回 x 和 y 中的最低值。

pow(x,y)：返回 x 的 y 次幂。

random()：返回 0 ～ 1 之间的随机数。

round(x)：把数四舍五入为最接近的整数。

sin(x)：返回数的正弦。

sqrt(x)：返回数的平方根。

tan(x)：返回角的正切。

toSource()：返回该对象的源代码。

valueOf()：返回 Math 对象的原始值。

(3) Date(日期)

Date 对象主要用于处理日期和时间。JavaScript 将与日期、时间相关的运算进行抽象、总结，并将其封装在 Date 中。但使用前必须使用 new 运算符创建一个实例，例如：

```
mydate = new Date();
```

Date 对象方法有：

getYear()：返回年份值。

getMonth()：返回月份值。

getDate()：并返回日期。

getDay()：返回星期几。

getHours()：返回小时数。

getMinutes()：返回分钟数。

getSeconds()：返回秒数。

getMilliseconds()：返回毫秒数。

getTime()：返回完整的时间。

setDate()：改变 Date 对象的日期。

setHours()：改变小时数。

setMinutes()：改变分钟数。

setMonth()：改变月份。

setSeconds()：改变秒数。

setTime()：改变完整的时间。

setYear()：改变年份。

4.1.7 内部对象系统

使用浏览器的内部对象系统，可实现与 HTML 文档进行交互。它的作用是将相关元素组织包装起来，提供给程序设计人员使用，从而减轻编程人的劳动，提高设计 Web 页面的能力。

常见的内部对象有：

窗口对象(window)：window 对象处于对象层次的最顶端，它提供了处理 navigator 窗口的方法和属性。

浏览器对象(navigator)：提供有关浏览器的信息。

位置对象(location)：location 对象提供了与当前打开的 URL 一起工作的方法和属性，它是一个静态的对象。

历史对象(history)：history 对象提供了与历史清单有关的信息。

文档对象(document)：document 对象包含了与文档元素一起工作的对象，它将这些元素封装起来供编程人员使用。

window 对象表示一个浏览器窗口或一个框架。在 JavaScript 中，window 对象是全局对象，所有的表达式都在当前的环境中计算。也就是说，要引用当前窗口根本不需要特殊的语法，可以把那个窗口的属性作为全局变量来使用。例如，可以只写 document，而不必写 window. document。同样，可以把当前窗口对象的方法当作函数来使用，如只写 alert()，而不必写 window. alert()。除了上面列出的属性和方法，window 对象还实现了核心 JavaScript 所定义的所有全局属性和方法。

下例是一个获取 WEB 浏览器信息的例子。例 4.10 的程序如下所示。

```
<html>
<head>
<title>获取 WEB 浏览器信息</title>
<body>
<script language = "JavaScript">
document.write("<h2>这是一个获取 Web 浏览器信息的程序</h2><br>");
document.write("浏览器名称: " + navigator.appName + "<br>");
document.write("版本号: " + navigator.appVersion + "<br>");
document.write("操作系统平台: " + navigator.platform + "<br>");
document.write("系统的 CPU 等级: " + navigator.cpuClass);
</script>
</body>
</html>
```

运行效果如图 4.13 所示。

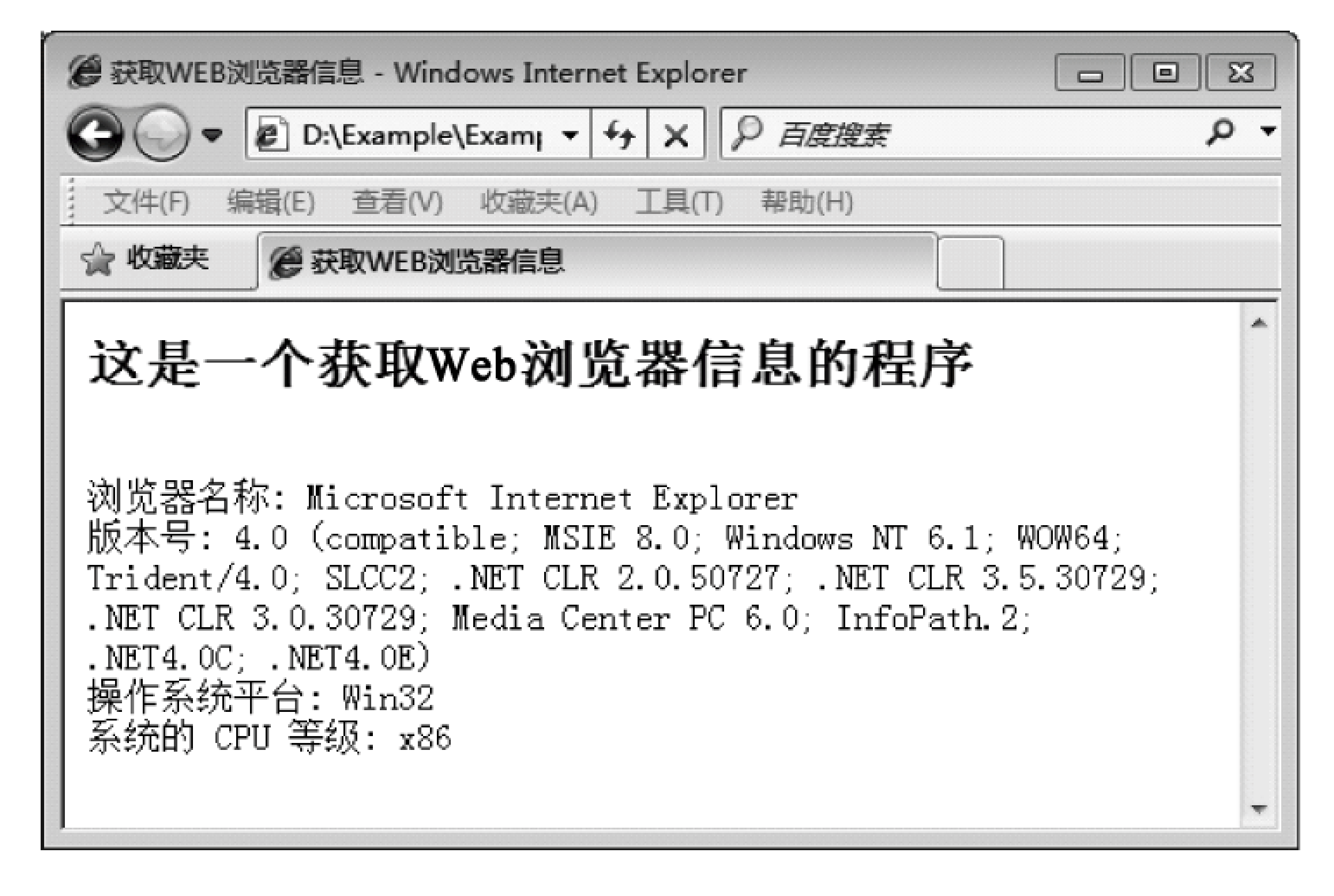


图 4.13 获取 Web 浏览器信息

JavaScript 中的输入可以通过窗口对象(window)来完成，输出可以通过文档对象(document)来完成。下例是一个关于输入输出的例子。例 4.12 的程序如下所示。

```
<Html>
<head>
<title>输入输出</title>
</head>
<body>
```

```
<script language="JavaScript">
var name;
name=window.prompt("请输入您的姓名：","张三");
document.write("尊敬的"+name+"：欢迎您访问我们的网页!");
</script>
</body>
</html>
```

运行效果如图 4.14 所示，将弹出一个对话框，提示用户输入姓名。

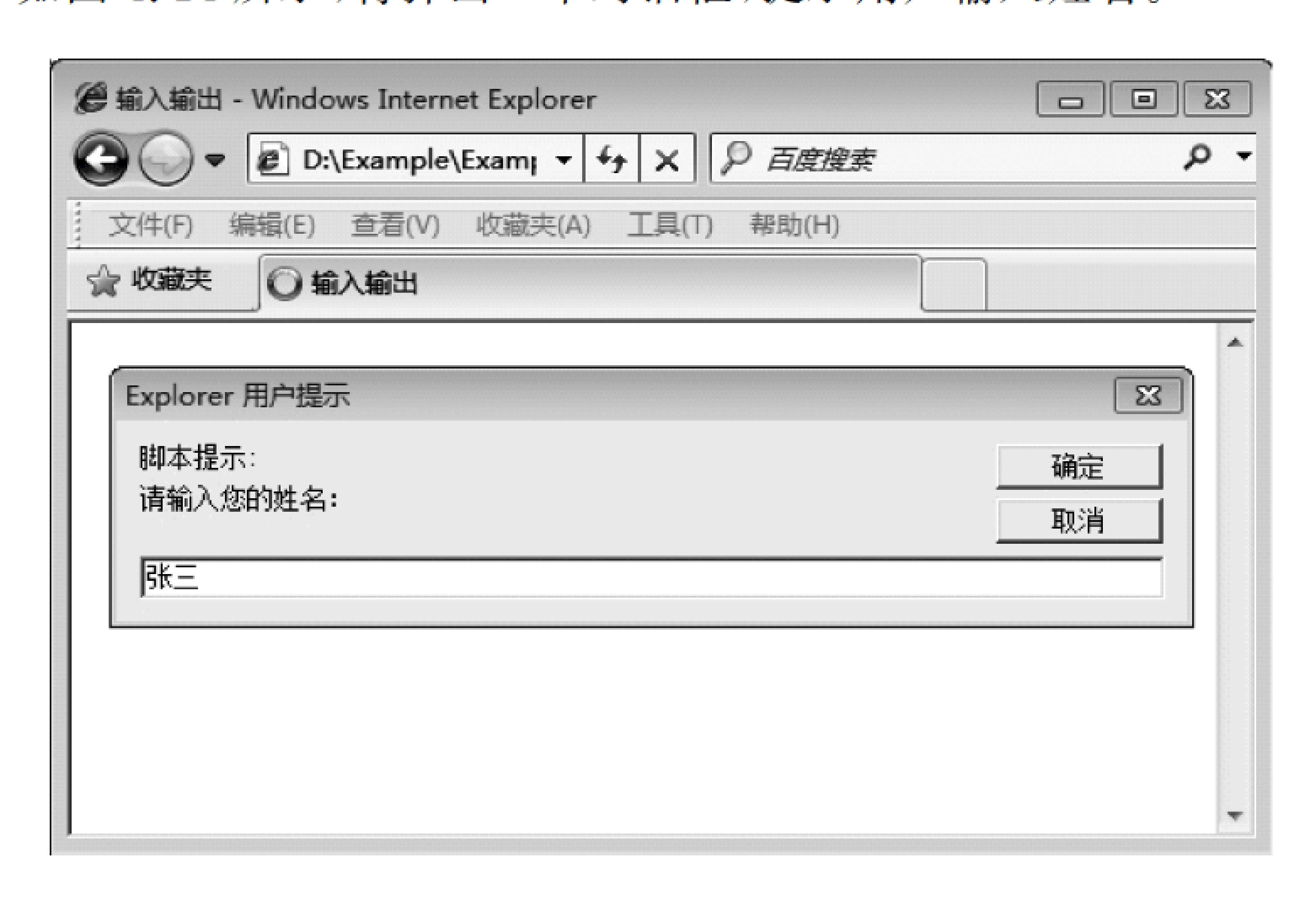


图 4.14 获取 Web 浏览器信息

当用户输入完姓名并单击“确定”按钮，在浏览器中将会根据用户输入显示相关信息。

提示： window.prompt 的第一个参数显示对话框提示信息，第二个参数是默认的输入信息。

新的顶层浏览器窗口由方法 window.open()创建。当调用该方法时，应把 open()调用的返回值存储在一个变量中，然后使用那个变量来引用新窗口。窗口的关闭可以用 close()函数来实现。

4.1.8 实例

1. 霓虹灯效果

下例是一个霓虹灯效果的例子，显示的文字如夜晚的霓虹灯一样被逐渐点亮，例 4.12 的程序如下所示。

```
<head>
<title>霓虹灯效果</title>
<script language="javascript">
    var str = "霓虹灯效果";
    var sLen = str.length;
    var i = 0;
    function neon() {
        i++;
```

```
        i = (i >= sLen)?0:i;
        i = (i < 0)?(sLen - 1):i;
        n.innerHTML = str.substring(0,i) + "<font color = '#FF0000>" + str.substring(i,
i + 1) + "</font>" + str.substring(i + 1,sLen);
    }
    setInterval("neon()",300);
</script>

</head>
<body>
<div id = "n" style = "font - size:50px;color:#FF9696"></div>
</body>
</html>
```

运行效果如图 4.15 示，显示的文字如夜幕下的霓虹灯一样逐渐一个一个亮起。

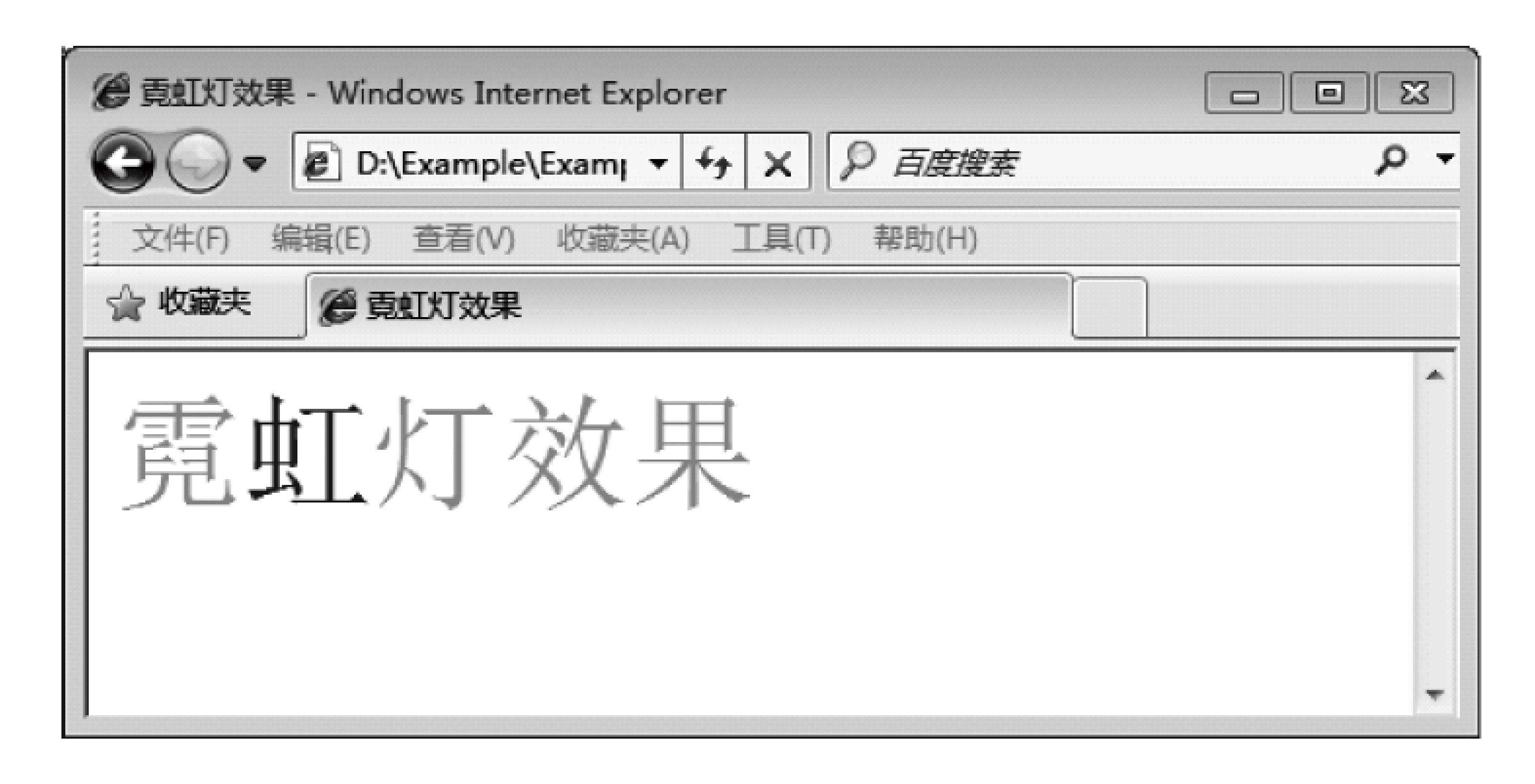


图 4.15 霓虹灯效果

2. 奔跑的文字

下例是一个奔跑文字效果的例子，例 4.13 的程序如下所示。

```
<html>
<head>
<title>奔跑的文字</title>
<style type = "text/css">
#content {position:absolute; left:0px; top:150px;
  width:100 % ; font - family:Impact; text - align:center;
  color:#336699; overflow:hidden;
}
</style>
<script language = JavaScript>
var s = 0;
var i = 0;
var m = 70;
var d = 2000;
var msg = new Array("Welcome to","Baidu","Search","Please","go!");
function changemessage(){
```

```
        if (i >= msg.length){
            location.href = ('http://www.baidu.com');
            return true;
        }
        txt = document.getElementById("content");
        txt.innerHTML = msg[i];
        s = 0;
        zoomTxt();
        i++;
        setTimeout("changemessage()", d);
    }
    function zoomTxt(){
        if(s < m){
            txt.style.fontSize = s;
            s += 5;
            setTimeout("zoomTxt()", 30);
        }
    }
    </script>
    </head>
    <body onload = "changemessage();">
    <div id = "content"></div>
    </body>
    </html>
```

运行效果如图 4.16 所示，显示的每组文字如动画一样运动起来，就像由远及近奔跑而来。

3. 时钟

下例是一个时钟效果的例子，在运行过程每隔一秒钟(即 1000 毫秒)显示的时间将会改变一次，例 4.14 的程序如下所示。

```
<html>
    <head>
        <title>时钟</title>
        <script language = JavaScript>
        function clock()
        { nowtime = new Date();
            document.all.timer1.innerText = "当前时间是:" + nowtime.getHours() + "时" +
nowtime.getMinutes() + "分" + nowtime.getSeconds() + "秒";
        }
        </script>
    </head>
    <body onload = "setInterval('clock()',1000)">
      <p id = "timer1"></p>
    </body>
</html>
```



图 4.16 奔跑的文字

运行效果如图 4.17 所示，在运行过程每隔一秒钟时间将会改变一次，从而达到时钟的效果。

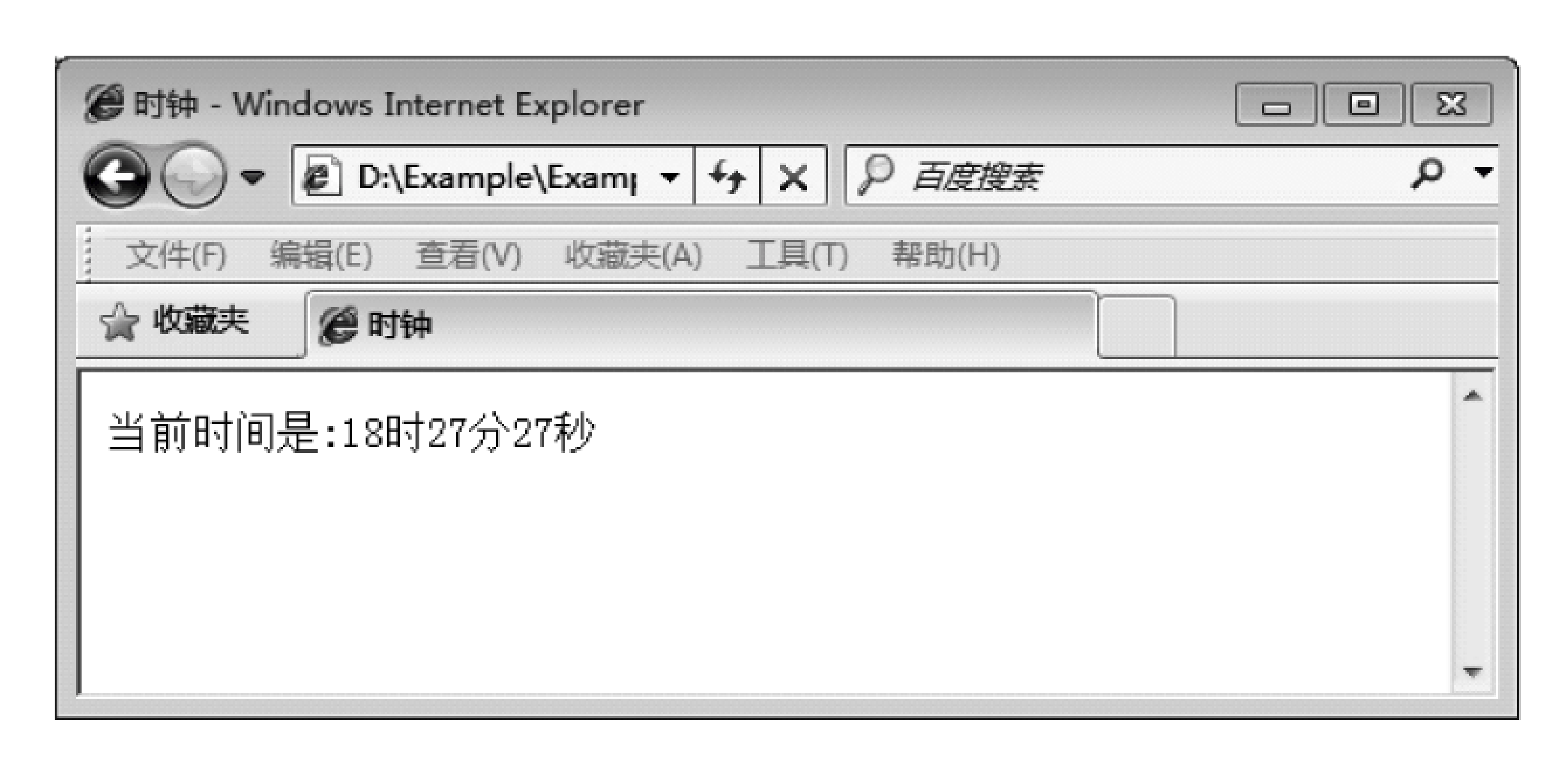


图 4.17 时钟

4.2 DOM 简介

文档对象模型(Document Object Model，DOM)，是 W3C 组织推荐的处理可扩展置标语言的应用程序编程接口(API)。

DOM 的历史可以追溯至 1990 年代后期微软与 Netscape 的“浏览器大战”(Browser Wars)，双方为了在 JavaScript 与 JScript 一决生死，于是大规模地赋予浏览器强大的功能。微软在网页技术上加入了不少专属事物，既有 VBScript、ActiveX，又有微软自家的

DHTML 格式等，使不少网页使用非微软平台及浏览器无法正常显示。DOM 即是当时酝酿出来的杰作。

提示： 应用程序接口(Application Programming Interface，API)，又称为应用编程接口，就是软件系统不同组成部分衔接的约定。API 主要目的是让应用程序开发人员得以调用一组例程功能，而无须考虑其底层的源代码为何或理解其内部工作机制的细节。由于近年来软件的规模日益庞大，常常需要把复杂的系统划分成小的组成部分，编程接口的设计十分重要。程序设计的实践中，编程接口的设计首先要使软件系统的职责得到合理划分。良好的接口设计可以降低系统各部分的相互依赖，提高组成单元的内聚性，降低组成单元间的耦合程度，从而提高系统的维护性和扩展性。

DOM 定义了访问 HTML 和 XML 文档的标准，它是中立于平台和语言的接口，它允许程序和脚本动态地访问和更新文档的内容、结构和样式。要改变页面的某个东西，JavaScript 就需要获得对 HTML 文档中所有元素进行访问的入口。这个入口，连同对 HTML 元素进行添加、移动、改变或移除的方法和属性，都是通过文档对象模型来获得的。

在 1998 年，W3C 发布了第一级的 DOM 规范，到目前为止共有 3 个 DOM 级别，分别是 DOM Level 1、DOM Level 2、DOM Level 3。

(1) DOM Level 1

在 1998 年 10 月份成为 W3C 的提议，由 DOM 核心与 DOM HTML 两个模块组成。DOM 核心能映射以 XML 为基础的文档结构，允许获取和操作文档的任意部分。DOM HTML 通过添加 HTML 专用的对象与函数对 DOM 核心进行了扩展。

(2) DOM Level 2

鉴于 DOM Level 1 仅以映射文档结构为目标，DOM Level 2 面向更为宽广。通过对原有 DOM 的扩展，DOM Level 2 通过对象接口增加了对鼠标和用户界面事件(DHTML 长期支持鼠标与用户界面事件)、范围、遍历(重复执行 DOM 文档)和层叠样式表(CSS)的支持。同时也对 DOM Level 1 的核心进行了扩展，从而可支持 XML 命名空间。

(3) DOM Level 3

DOM Level 3 通过引入统一方式载入和保存文档和文档验证方法对 DOM 进行进一步扩展，DOM Level 3 包含一个名为“DOM 载入与保存”的新模块，DOM 核心扩展后可支持 XML1.0 的所有内容，包括 XML Infoset、XPath、和 XML Base。

当阅读与 DOM 有关的材料时，可能会遇到参考 DOM Level 0 的情况。需要注意的是并没有标准被称为 DOM Level 0，它仅是 DOM 历史上一个参考点(DOM Level 0 被认为是在 Internet Explorer 4.0 与 Netscape Navigator4.0 支持的最早的 DHTML)。

DOM 的优势主要表现在：易用性强，使用 DOM 时，将把所有的 XML 文档信息都存于内存中，并且遍历简单，支持 XPath，增强了易用性。

DOM 的缺点主要表现在：效率低，解析速度慢，内存占用量过高，对于大文件来说几乎不可能使用。另外效率低还表现在大量地消耗时间，因为使用 DOM 进行解析时，将为文档的每个 element、attribute、processing-instrUCtion 和 comment 都创建一个对象，这样在

DOM 机制中所运用的大量对象的创建和销毁无疑会影响其效率。

总之，由于 DOM 独立于语言、浏览器和平台，因此使得 web 开发者通过 JavaScript 能够轻松地访问和操作 HTML 文档中的元素，从而轻松使开发出来的页面动态效果更加灵活和人性化。

4.3 jQuery 基础

4.3.1 初识 jQuery

jQuery 是最初由美国人 John Resig 于 2006 年初创建的开源 JavaScript 项目，后来由于 jQuery 优秀的设计以及开源属性，吸引了来自世界各地的众多高手加入其中。

jQuery 的宗旨是 write less, do more(写的更少，做的更多)，下面通过一些实例进行说明 jQuery 是如何做到的。

例如要取得一个 DOM 对象。

传统的 JavaScript 代码如下：

```
var p1 = document.getElementById("p1");
```

jQuery 代码如下：

```
var p1 = $('#feed');
```

再例如例 4.1 中，首先由写如下代码实现 welcome()函数：

```
<script language = JavaScript>
    function welcome() {
    alert("欢迎学习 JavaScript 语言!");}
</script>
```

然后要在 HTML 主程序中加入下面 HTML 代码：

```
<a href = "#" onclick = "welcome();">欢迎进入 JavaScript 世界!</a>
```

而 jQuery 的写法如下：

```
<script>
  $("a").click(function() {
   alert("欢迎学习 JavaScript 语言!");});
</script>
```

HTML 中调用 jQuery 代码如下：

```
<a href = "#">欢迎进入 JavaScript 世界!</a>
```

结果一样，但相比较而言明显后者显得更加简洁、易懂。

jQuery 还有完整丰富的在线学习联机帮助教程，如图 4.18 所示，可参考 http://learn.jquery.com/。



图 4.18　jQuery 在线学习教程

4.3.2　搭建 jQuery 运行环境

读者可以从 jQuery 的官方网站(http://jquery.com/download/#Download_jQuery,如图 4.19 所示)下载 jQuery 的任何一个版本,目前最新版本是 jQuery2.1.1。jQuery2.x 的 API 和 jQuery1.x 基本相同,但 jQuery2.x 不支持 Internet Explorer 6,Internet Explorer 7 或 Internet Explorer 8。鉴于目前 Internet Explorer 8 比较流行,这里建议读者先使用 jQuery1.x。

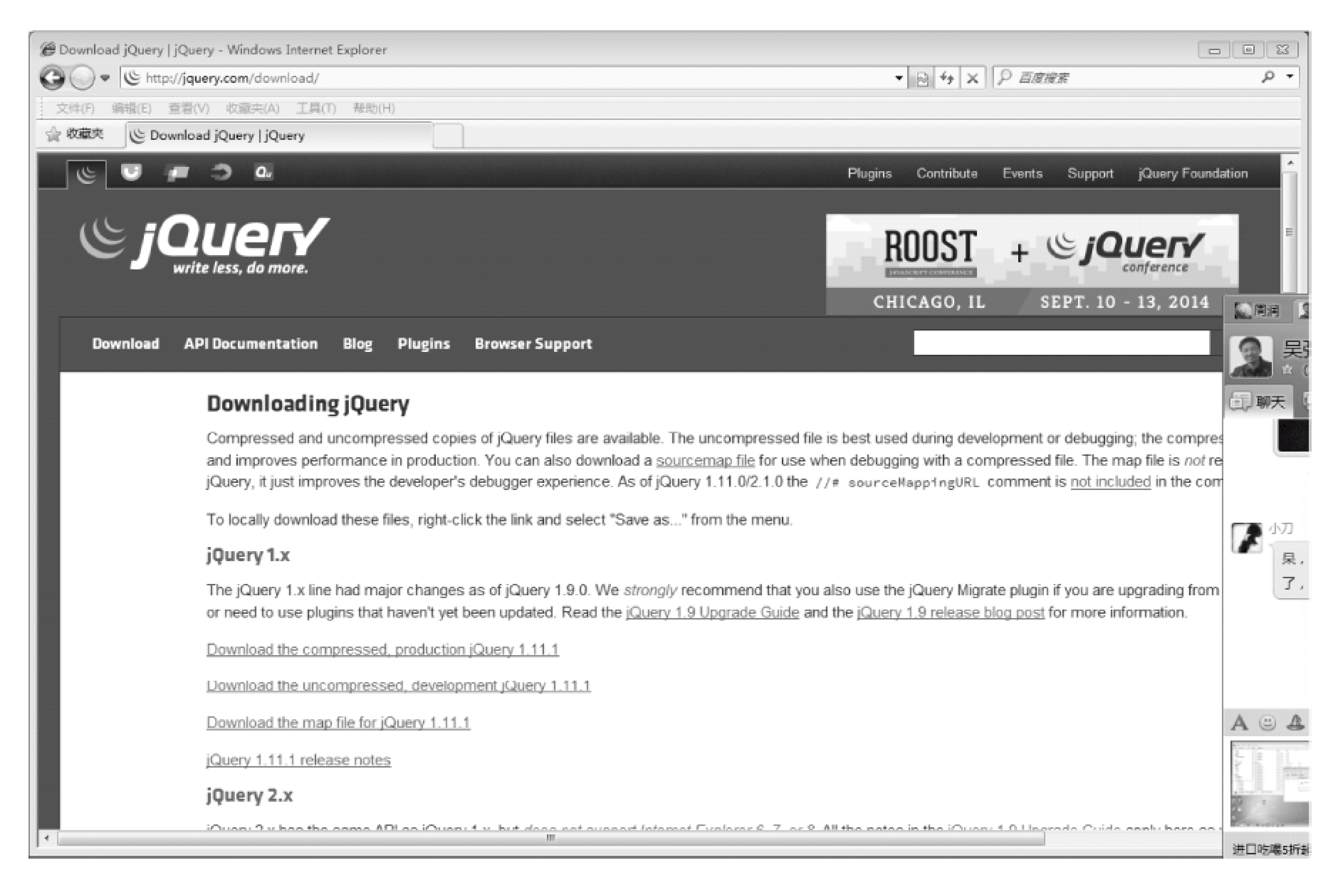


图 4.19　下载 jQuery

从 jQuery 的官方网站下载 jQuery 库后，编程人员就可以使用编程了(本书使用的是 jquery-1.11.1.js)。

下面的例 4.15 是一个使用 jQuery 编写的简单示例，视图界面是由一个文本框和一个命令按钮组成。在文本框中输入名字后，单击命令按钮将弹出一个对话框，显示欢迎界面，如图 4.20 所示。

```
<html>
<head>
<title>欢迎学习 jQuery</title>
<script type = "text/JavaScript" src = "JS/jquery - 1.11.1.js"> </script>
<script type = "text/JavaScript">
    $ (document).ready(function() {
      $ ("input[type = 'submit']").click(function() {
        var msg = $ ("input[name = 'name']").val() + ",欢迎进入 jQuery 世界!";
        alert(msg);
      });
    });
  </script>
  </head>
  <body>
    <input name = "name" /> <br />
    <input type = "submit" value = "欢迎">
  </body>
  </html>
```

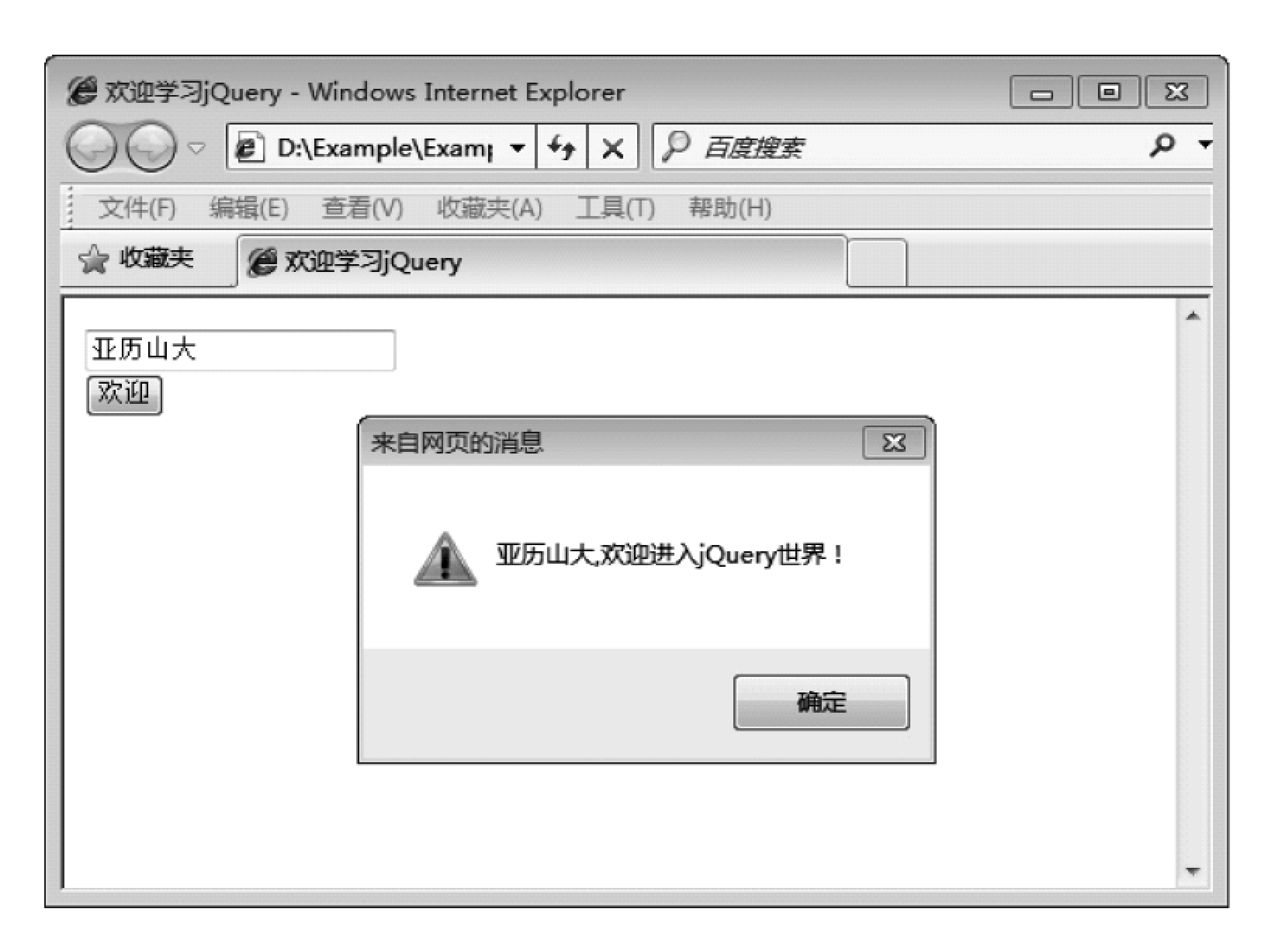


图 4.20 jQuery 示例

本例中引用了 JS 目录下的 jQuery 库 jquery-1.11.1.js，再对命令按钮绑定单击处理函数，当输入信息后单击命令按钮时会调用处理函数，弹出欢迎对话框。

4.3.3 jQuery 实战开发与应用

jQuery 是一个轻量级的库，拥有简洁的语法、优雅的代码风格、强大的选择器、可靠的事件处理机制、出色的 DOM 操作、完善的跨平台兼容性等优点，从而吸引了一批又一批的优秀 JavaSript 开发人员转投其中。同时 jQuery 又是一个开源产品，透明、高质量、低风险。

1. jQuery 核心函数

在 jQuery 中，所有的 DOM 对象都将封装成 jQuery 对象，而且只有 jQquery 对象才能使用 jQuery 的方法或者属性来执行相关的操作。所以，jQuery 提供了一个可以将 DOM 对象封装成 jQuery 对象的函数，就是 jQuery 核心函数 jQuery，也称为工厂函数。下面将介绍 jQuery 核心函数，在 jQuery 编程中将起到至关重要的作用，而在实际开发中用到最多的也是它们。

jQuery 核心函数有一个非常简单的别名 $（美元符号），调用 jQuery 核心函数时都可以写成 $([arguements])。

- jQuery()。该函数返回一个空的 jQuery 对象。
- jQuery(elements)。该函数将一个或者多个 DOM 元素转化为 jQuery 对象，jQuery(document)这个函数也可以将 XML 文档和 window 对象转化为有效的参数。
- jQuery(callback)。该函数是 jQuery(document).ready(callback)的简写，该函数将绑定一个在 DOM 文档加载完成后执行的函数。页面中所有需要在 DOM 加载完成时执行的 jQuery 操作都需要包含在这个函数中。
- jQuery(html)。该函数根据提供的 HTML 标记代码，动态创建由 jQuery 对象封装的 DOM 元素，如 jQuery("<div></div>")。
- jQuery(html, props)。该函数根据提供的 HTML 标记代码，动态创建由 jQuery 对象封装的 DOM 元素，同时对该 DOM 元素设置一组属性、事件等。
- jQuery(html, [owerDocument])。该函数根据提供的 HTML 标记代码，动态创建由 jQuery 对象封装的 DOM 元素，并且制定该 DOM 元素所在的文档。
- jQuery(expression, [context])。该函数接受一个包含 jQuery 选择器的字符串，然后用这个字符串去匹配一个或多个元素。jQuery 的核心功能都是通过这个函数实现的。

2. jQuery 选择器

jQuery 选择器是 jQuery 最核心的部分，也是其根基。所有 DOM 操作、事件操作、Ajax 操作都离不开选择器。要想操作某 HTML 元素，首先要准确找到该 HTML 元素。熟练使用 jQuery 选择器可以在很大程度上简化开发人员的编程工作。

jQuery 选择器可以分为以下四类：

(1) 基本选择器(表 4.6)

基本选择器是 jQuery 选择器中最基本也是最常用的选择器。基本选择器有 5 种，即标签选择器、ID 选择器、类选择器、通用选择器、群组选择器。基本选择器通过元素 id、class、标签名等来查找 DOM 元素。在网页中，每个 id 名称只能用一次，而 class 允许重复使用。

表 4.6 基本选择器

选择器	描述	语法
标签选择器	用于选择页面中已有的标签元素	$("element")
ID 选择器	用于获取某个具有 id 属性的元素	$("id")
类选择器	用于获取某个具有 class 属性的元素	$("class")
通用选择器	匹配所有元素	$("*")
群组选择器	用于选择所有指定的选择器组合的结果	$("selector1,…,selectorN")

(2) 层次选择器(表 4.7)

在 HTML 文档中,每个元素总是处于 DOM 节点树的某个位置,文档层次结构中元素之间总是存在某种层次关系,比如父子级关系等,在 jQuery 中,可以使用层次选择器来获取相关元素。

表 4.7 层次选择器

选择器	描述	语法
子元素选择器	用于给定父元素下查找其所有子元素	$("parent>child")
后代选择器	用于给定的祖先元素下匹配所有后代元素	$("ancestor descendant")
紧邻同辈选择器	用于匹配紧接在 prev 元素后的 next 元素	$("prev+next")
相邻同辈选择器	用于选择某元素后的所有同辈元素	$("prev~siblings")

(3) 表单域选择器(表 4.8)

为了使编程人员能够更加灵活地操作表单,jQuery 加入了表单域选择器。通过表单域选择器,编程人员能够非常灵活地获取表单中的某个或某类元素。这组选择器都以一个冒号(:)开始,大多数可独立使用。

表 4.8 表单域选择器

选择器	描述	语法
:input 选择器	用于选择 input、textarea、select、button 元素	$(":input")
:text 选择器	用于选择单行文本框元素	$(":text")
:password 选择器	用于选择密码框元素	$(":password ")
:radio 选择器	用于选择单选按钮元素	$(":radio ")
:checkbox 选择器	用于选择复选框元素	$(":checkbox ")
:file 选择器	用于选择文件域元素	$(":file ")
:image 选择器	用于选择图像域元素	$(":image ")
:hidden 选择器	用于选择不可见元素以及隐藏域元素	$(":hidden ")
:button 选择器	用于选择按钮元素	$(":button ")
:submit 选择器	用于选择提交按钮元素	$(":submit ")
:reset 选择器	用于选择重置按钮元素	$(":reset ")

(4) 过滤选择器

过滤选择器是通过特定的过滤规则来筛选所需的 DOM 元素。在使用基本选择器时,jQuery 对象通常会包含一组 DOM 对象。而在实际应用中,通常还要根据特定条件从中筛选出一部分 DOM 元素。这种情况下,可以在基本选择器的基础上通过添加过滤选择器来

完成筛选功能。过滤规则与 CSS 中的伪类选择器语法相同，即选择器都以一个冒号(：)开始。根据过滤规则，过滤选择器可以分为基本过滤、内容过滤、可见性过滤、属性过滤、子元素过滤以及表单对象属性过滤。

4. jQuery 的 DOM 操作

DOM 操作可分为 DOM Core(核心)、HTML-DOM 和 CSS-DOM。DOM Core 并不专属于 JavaScript，任何一种支持 DOM 的程序设计语言都可以使用。DOM Core 的用途也并非仅限于处理网页，也可以用来处理任何一种使用标记语言写的文档，例如 XML。HTML-DOM 主要用于提供一些简明的记号来描述各种 HTML 元素的属性。而 CSS-DOM 是针对 CSS 的操作，其主要作用是获取和设置 style 对象的各种属性，并通过改变其属性值使网页呈现各种不同的效果。

在动态改变 HTML 文档内容的过程中，操作文档中的节点是最常见、最频繁的操作之一。如果能顺利实现这些操作，将能够灵活快捷地实现任何想要的页面效果。

下面介绍 jQuery 的 DOM 操作。首先介绍一下思路，每一张网页都可以用 DOM 表示出来，而每一份 DOM 都可以被看作一棵 DOM 树，对 DOM 的各种操作都围绕这棵 DOM 树展开。通过 jQuery 选择器可以非常容易地在 DOM 树上查找元素节点。当找到所需元素后，可使用 attr()方法来获取其各种属性值。当然，也可以轻松实现对元素节点的删除、复制、替换、遍历等操作。

5. jQuery 的事件

事件是脚本编程的灵魂。JavaScript 是事件驱动的脚本编程语言。传统的 JavaScript 中，可以把事件分为鼠标事件、键盘事件、页面事件、表单事件等。jQuery 对 JavaScript 中的常用事件进行封装。通过对 JavaScript 事件的封装，jQuery 消除了各种浏览器之间的差异，使编程人员编程时更加方便，代码兼容性更好。

(1) DOM 载入事件

在 JavaScript 中，通常使用 window. onload 方法，而在 jQuery 中提供了一个 DOM 加载完成事件，即 ready 事件。ready 事件是一个可以在 DOM 加载完成后执行的事件，它可以绑定多个响应函数。ready 事件可大大提高 Web 页面响应速度，是 jQuery 事件模块中最重要的事件之一。

ready 事件对所有 jQuery 对象都有效，可以绑定到所有被 jQuery 对象封装过的 DOM 元素上。比如为页面 document 对象、window 对象、html 节点、body 节点、DOM 元素 div、input 等分别进行 document 对象的绑定，在绑定事件处理程序中可以分别向页面输出不同的执行。例如 document 对象的 ready 事件可以用如下代码：

```
$ (document).ready(function() {
  //编写代码
})
```

(2) 事件绑定

当文档加载完毕后，可以使用 bind()方法对匹配元素进行事件的绑定来完成某些操作。bind()方法的语法结构如下：

```
bind(type,[data],fn)
```

其中第一个参数是事件类型,类型包括:focus、load、unload、resize、scroll、click、dbclick、mousedown、mouseup、mousemove、mouseover、select、submit、keydown、keypress 等,也可以是自定义名称。

第二个参数是可选参数,作为 event.data 属性值传递给事件对象的额外数据对象。

第三个参数是用来绑定的处理函数。

(3) 合成事件

jQuery 有两个合成事件:hover()方法和 toggle()方法,这两个方法都属于 jQuery 自定义的方法。

hover()方法的语法结构如下:

```
hover(enter, leave);
```

hover()方法用于模拟光标悬停事件。当光标移动到元素上时,会触发第一个函数 enter;当光标移出该元素时,会触发第二个函数 leave。

toggle()方法的语法结构如下:

```
toggle (fn1,fn2, …, fnN);
```

toggle()方法用于模拟鼠标连续单击事件。第一次单击时,触发第一个函数 fn1;第二次单击时,触发第二个函数 fn2;以此类推。

(4) 模拟操作

很多事件是用户必须进行操作才能触发,例如单击按钮可触发 click 事件,但有时需通过模拟用户操作达到触发事件的效果。在 jQuery 中,可以使用 trigger()方法模拟操作。例如如果想触发 id 为 button1 按钮的 click 事件,可以使用如下代码:

```
$('#button1').trigger("click");
```

当浏览器运行到该代码时,就立刻输出单击按钮的效果了。

trigger()方法不仅能触发浏览器支持的具有相同名称的事件,也可以触发自定义名称的事件。trigger()方法的语法结构如下:

```
trigger(type,[data]);
```

其中第一个参数是要触发的事件;第二个参数是可选参数,是要传递给事件处理函数的附件数据,以数组形式传递。

6. jQuery 的动画

动画效果是 jQuery 最吸引人的地方。通过 jQuery 的动画方法,可以轻松为网页添加非常精彩的视觉效果,给用户一种全新的体验。

(1) 基本效果

jQuery 动画中的基本效果有隐藏、显示和交替隐藏显示。

hide()方法可将被选中元素隐藏,hide()方法的语法结构如下:

```
hide(speed,[callback]);
```

hide()方法可根据指定的速度 speed(单位：毫秒)动态地改变匹配元素的高度、宽度和不透明度,从而以优雅的动画隐藏匹配元素。

下面通过例 4.16 来说明,当单击图片时,图片会以优雅的动画慢慢隐藏消失。如图 4.21 所示。

```
<html>
        <head>
                <title>动态隐藏动画</title>
        <script type="text/JavaScript" src="JS/jquery-1.11.1.js"></script>
        <script type="text/JavaScript">
          $(document).ready(function() {
              $("img").click(function(){
                $(this).hide(7000);
              })
          })
        </script>
        </head>
        <body>
        <div><img src="night.bmp" /></div>
        </body>
</html>
```

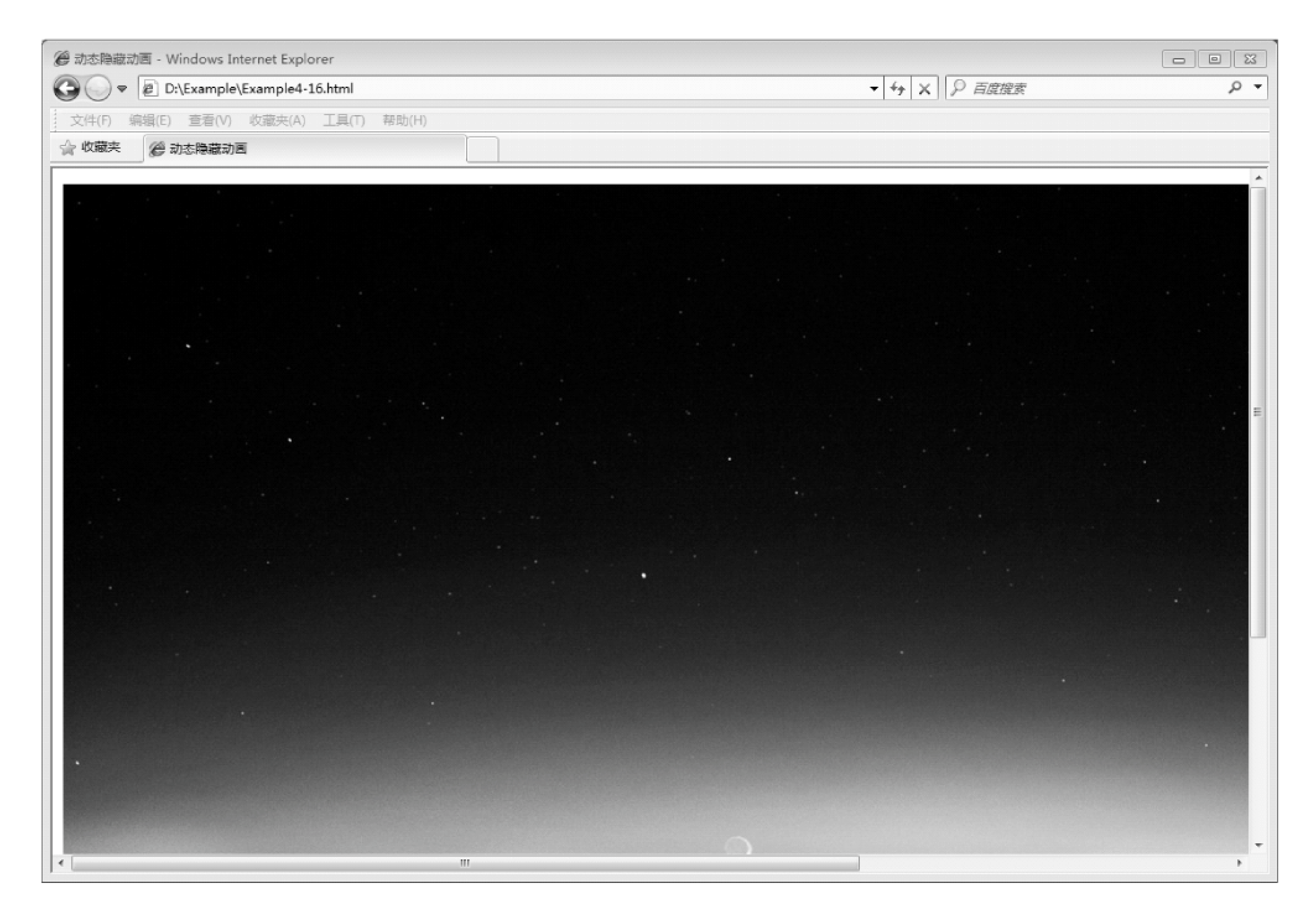


图 4.21　jQuery 动态隐藏动画

show()方法可将被选中的隐藏的元素显示出来,show()方法的语法结构如下:

```
show(speed,[callback]);
```

同 hide()方法一样,show()方法可根据指定的速度 speed(单位：毫秒)动态地改变匹

配元素的高度、宽度和不透明度，从而以优雅的动画将隐藏的匹配元素显示出来。

toggle()方法可切换被选中的元素的可见状态，toggle()方法的语法结构如下：

```
toggle (speed,[callback]);
```

toggle()方法可参考 hide()方法。

(2) 滑动效果

jQuery 动画中的滑动效果包括向上收缩、向下展开和交替伸缩。

slideUp()方法可通过调整元素高度(向上减小)来动态隐藏所匹配的元素，slideUp()方法的语法结构如下：

```
slideUp((speed,[callback]);
```

第一个参数 speed 的单位是毫秒，该动画效果只调整元素的高度；第二个参数 callback 是可选参数，表示在动画完成时执行的函数。

下面通过例 4.17 来说明，当单击图片时，图片会以优雅的动画慢慢向上收缩直至消失。如图 4.22 所示。

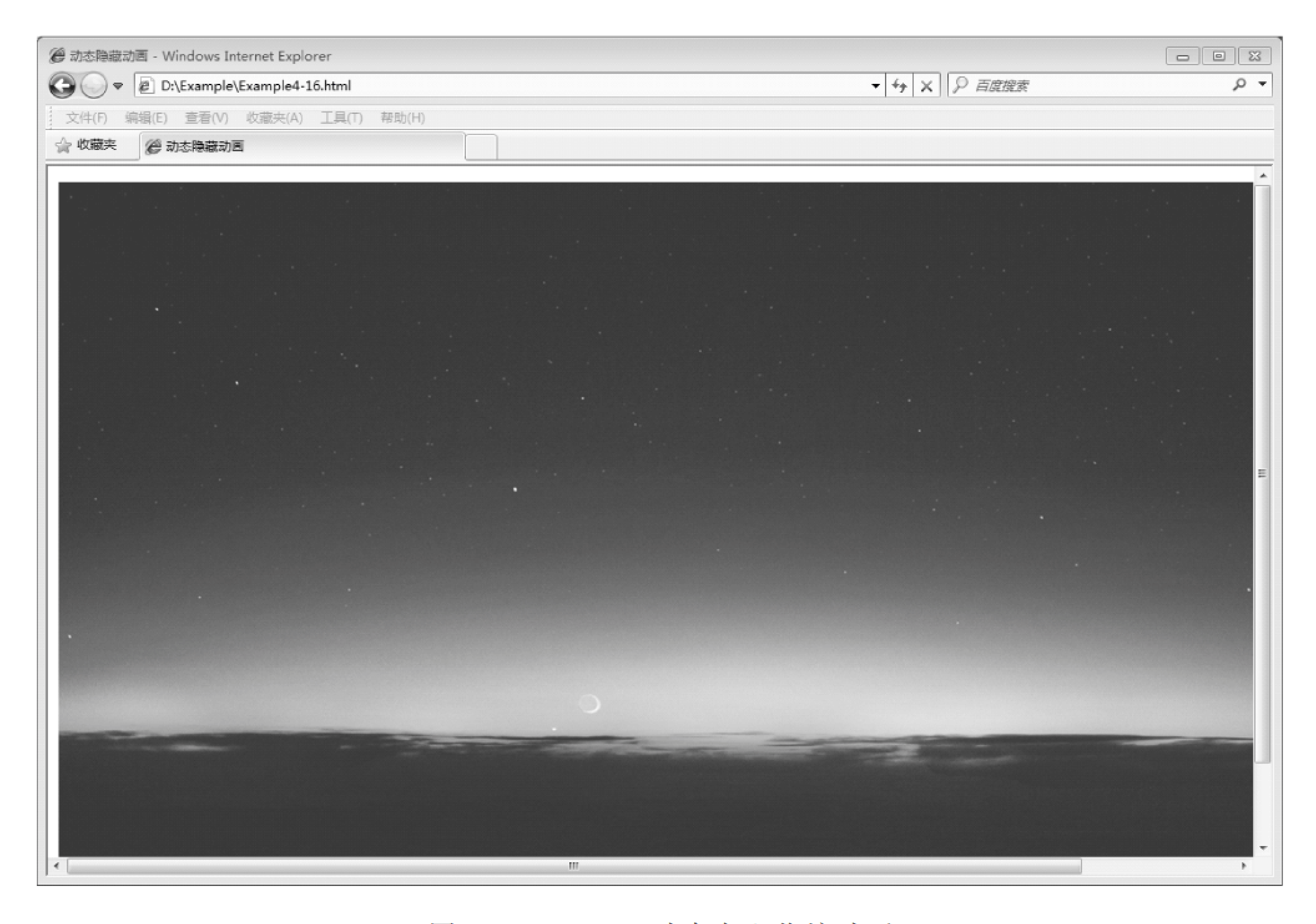


图 4.22 jQuery 动态向上收缩动画

```
<html>
        <head>
                <title>动态向上收缩动画</title>
       <script type="text/JavaScript" src="JS/jquery-1.11.1.js">

</script>
      <script type="text/JavaScript">
        $(document).ready(function() {
```

```
            $ ("img").click(function(){
              $ (this).slideUp(7000);
            })
        })
        </script>
        </head>
        <body>
        <div><img src = "night.bmp" /></div>
        </body>
</html>
```

slideDown()方法可通过调整元素高度(向上减小)来动态隐藏所匹配的元素,slideDown()方法的语法结构如下:

```
slideDown ((speed,[callback]);
```

slideDown()方法的参数适用可参照 slideUp ()方法。

slideToggle()方法可通过高度变化切换被匹配的元素的可见状态,toggle()方法的语法结构如下:

```
slideToggle (speed,[callback]);
```

slideToggle()方法的参数适用可参照 slideUp()方法。

(3) 淡入淡出效果

淡入淡出效果是许多网站上最常见的图片动画效果。jQuery 动画中的淡入淡出效果包括淡入效果、淡出效果和自定义不透明度。

fadeIn()方法可通过不透明度的变化来实现所匹配元素的淡入效果,fadeIn()方法的语法结构如下:

```
fadeIn ((speed,[callback]);
```

同样,第一个参数 speed 的单位是毫秒,该动画效果只调整元素的高度;第二个参数 callback 是可选参数,表示在动画完成时执行的函数。

下面通过例 4.18 来说明,当页面被载入时,图片会在 7 秒内从无到有慢慢显示出来。如图 4.23 所示。

```
<html>
        <head>
                <title>淡入动画</title>
       <script type = "text/JavaScript" src = "JS/jquery-1.11.1.js">

</script>
      <script type = "text/JavaScript">
        $ (document).ready(function() {
            $ ("div").fadeIn(7000);
            })
        </script>
        </head>
        <body>
        <div style = "display:none;"><img src = "night.bmp" /></div>
```

```
    </body>
</html>
```

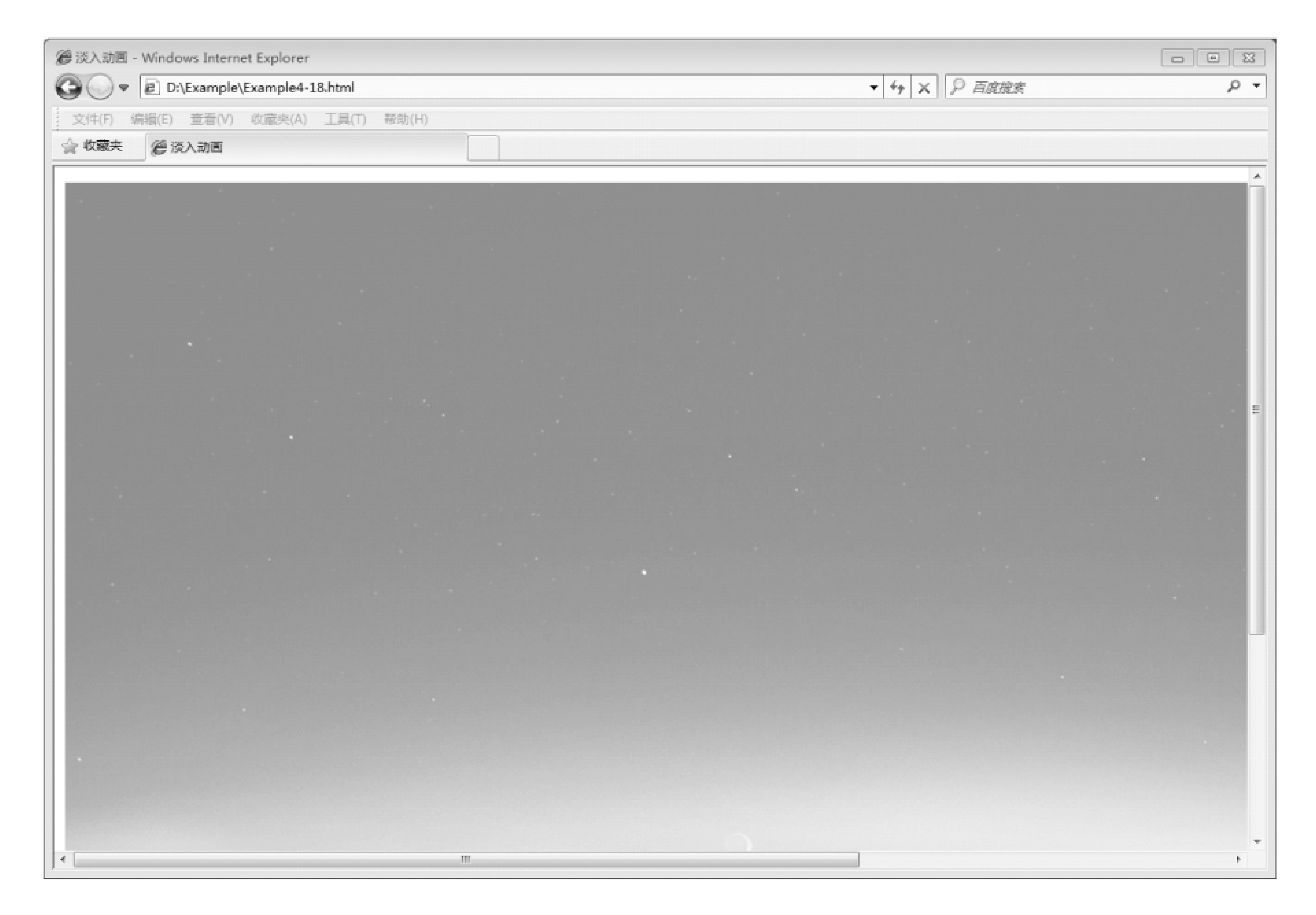


图 4.23 jQuery 淡入动画

fadeOut()方法可通过不透明度的变化来实现所匹配元素的淡出效果，fadeOut()方法的语法结构如下：

```
fadeOut ((speed,[callback]);
```

fadeOut()方法的参数适用可参照 fadeIn ()方法。

fadeTo()方法可通过不透明度的渐进变化来将所匹配元素调整到指定的不透明度，fadeTo()方法的语法结构如下：

```
fadeTo ((speed, opacity ,[callback]);
```

fadeTo()方法的第二个参数 opacity 表示要调整道德不透明度值(0～1,1 表示全部显示)，其余参数适用可参照 fadeIn()方法。

4.4 Ajax

Ajax 指异步 JavaScript 及 XML(Asynchronous JavaScript And XML)，它并不是指某一种单一的技术，而是一系列网页相关应用相关技术的有机结合体。Ajax 能够实现在网页页面无刷新的情况下从服务器端获取数据，可以极大程度上改善用户体验。

Ajax 技术早在 20 世纪 90 年代末就已经在浏览器中出现，但当时不是称为 Ajax，最近几年才被人们所重视。最初，Ajax 技术只是在 IE 浏览器中得到支持，后来其他浏览器如 Firefox、Safari 等都开始支持 Ajax 技术，慢慢地开始普及并流行开了。

Ajax 基于下列 Web 标准：

(1) JavaScript

(2) XML

(3) HTML

(4) CSS

(5) DOM

目前，在 Ajax 中使用的 Web 标准已被良好定义，并被几乎所有的主流浏览器支持，如 IE、Firefox、Safari、Chome、Opera 等。Ajax 应用程序独立于浏览器和平台。在 2005 年 Ajax 被 Google 推广开来(Google Suggest)，Orkut、Gmail、Google Groups、Google Maps、Google Suggest 都应用了这项技术。

这里先讨论一下两种典型的客户端实现方式：

(1) 瘦客户端。完全依赖服务器计算，客户端仅仅显示服务器数据。典型例子如 Web 应用中的浏览器，浏览器仅负责显示逻辑控制，客户端计算基本闲置，所有页面均由服务器生成。

(2) 胖客户端。客户端承担一部分计算工作，合理利用客户端资源，分担服务器计算压力。Ajax 和 Flash 是现在胖客户端的主流。

随着 Internet 的高速发展，当前 Internet 上的用户和数据量都呈现爆炸性增长，而同时服务器端面临的压力也随之增大，网络上传输的数据也呈现爆炸性增长；虽然计算机及网络硬件技术也比以前有所改进，但仅通过提升服务器硬件和提高网络带宽来适应用户和数据量的爆炸性增长是不太现实的，一方面硬件技术的提升速度远远无法满足用户和数据量的增加，另一方面硬件的提升将会伴随着成本的增加，在有限的成本内通过提升硬件来满足用户和数据量的增加几乎是无法想象的。

Ajax 技术不是一种新的编程语言，而是一种用于创建更好更快以及交互性更强的 Web 应用程序的技术。Ajax 技术的核心是 JavaScript 的 XMLHttpRequest 对象，它是一种支持异步请求的技术。通过使用 JavaScript 的 XMLHttpRequest 对象来直接与服务器提出请求并处理响应，JavaScript 可在不重载页面的情况下与 Web 服务器交换数据。Ajax 技术在浏览器与 Web 服务器之间使用异步数据传输(HTTP 请求)，这样就可使网页从服务器请求少量的信息，而不是整个页面。Ajax 可使因特网应用程序更小、更快、更友好。

在传统的 JavaScript 编程中，用户如果希望从服务器上的文件或数据库中得到任何信息，或者向服务器发送信息的话，就必须利用一个 HTML 表单向服务器 GET 或 POST 数据。而用户则需要单击“提交”按钮来发送/获取信息，等待服务器的响应，然后刷新一张新的页面。这个过程，页面可能一片空白，用户只能等待。

通过利用 Ajax，通过 JavaScript 的 XMLHttpRequest 对象，直接与服务器通信。Ajax 使用 XML 传输数据、异步 JavaScript 解析 XMLDocument 对象，服务器回传数据时不重绘页面，网络传输时耗可以同时降低，而且 Ajax 可以与任何服务器技术进行交互。但需要强调的是 Ajax 是典型的客户端技术。

Ajax 对浏览器有一定依赖，任何对 Ajax 的应用都建立在浏览器基础上。通过使用 HTTP 请求，WEB 页可向服务器发送请求，并得到来自服务器的响应，而不加载页面。用户可以停留在同一个页面，而不会注意到脚本在后台请求过页面，或向服务器发送过数据。

```
<html>
```

```
<head>
<title>Ajax</title>

<script language = JavaScript>

function AjaxFunction()
{
var xmlHttp;
try{
    xmlHttp = new XMLHttpRequest();
    }
catch (e){
try{
xmlHttp = new ActiveXObject("Msxml2.XMLHTTP");
  }
   catch (e){
   try{
    xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
      }
      catch (e)
         {
         alert("Your browser can not run Ajax!");
         return false;
         }
      }
    }
    xmlHttp.onreadystatechange = function()
      {
      if(xmlHttp.readyState == 4)
        {
         document.myForm.time.value = xmlHttp.responseText;
        }
      }
    xmlHttp.open("GET","servertime.asp",true);
    xmlHttp.send(null);

}
</script>
</head>
<body>
<form name = "myForm">
用户的姓名: <input type = "text" name = "username" onkeyup = "AjaxFunction();"><br>
服务器时间: <input type = "text" name = "time">
</form>

</body>
</html>
```

下面编写服务器端程序“servertime.asp”:

```
<%
```

```
response.expires = -1
response.write(time)
%>
```

当用户在姓名文本框中输入姓名时，在时间文本框中可在不加载页面的情况下利用servertime.asp获得服务器的时间。

jQuery中的Ajax对JavaScript中的Ajax进行了封装，jQuery中的Ajax实现方法最大的一个优势就是消除了各种浏览器之间的差异，提高了兼容性，从而编程人员不用再关心程序对浏览器的兼容性，其次也简化了编程人员的工作，不必编写大量代码。

jQuery对JavaScript中的Ajax进行了3层封装：第1层封装并实现了$.ajax()方法；第2层封装并实现了load()方法、$.get()方法和$.post()方法；第3层封装并实现了$.getScript()方法和$.getJSON()方法。

jQuery中Ajax的优势还在于对整个请求响应过程能够进行全程监控，为此jQuery中提供了一系列全局事件函数，能够捕获所有匹配Ajax事件并注册回调函数。这样，当请求数据量较大或服务器比较繁忙时，页面可能长时间无反应，用户可能会无所适从，这时可在页面中给用户一个提示，以减少用户的焦虑感，增强人机交互的人性化。

世界上没有完美的事物，Ajax也不是一种完美的技术，jQuery的出现有效地补充了Ajax技术的不足之处。

小结

本章主要介绍JavaScript和jQuery中的基本知识，并简单介绍了Ajax技术。JavaScript语言是基础，而jQuery是其上非常耀眼的一朵鲜花。jQuery是最近几年非常流利、非常火的一种JavaScript库，甚至还出现了基于jQuery框架在移动设备上应用的jQuery Mobile项目。

本章从介绍JavaScript的基本知识入手，讲解了数据类型、操作符、流程控制以及命名空间的声明等内容。并且通过一些具体的实例，让读者了解开发JavaScript和jQuery程序的过程。Ajax并不是什么新鲜事物，作为客户端技术，Ajax利用异步服务器交互、服务器回传数据不重绘页面，可以减轻服务器端压力和网络传输时耗。但Ajax技术不能滥用，否则可能会适得其反。

习题

1. JavaScript语言最初是由________创建的，一开始命名是________。

2. JavaScript语言的基本数据类型有________、________、________、________。

3. jQuery的宗旨是____________________。

4. Ajax的全称是____________________。

5. 我国古代数学家张丘建在《算经》中提出了著名的“百钱百鸡问题”：“鸡翁一，值钱五；鸡母一，值钱三；鸡雏三，值钱一；百钱买百鸡，翁、母、雏各几何?”意思是说：一只公鸡

卖 5 枚钱，一只母鸡卖 3 枚钱，三只小鸡卖 1 枚钱，用 100 枚钱买 100 只鸡，能买到公鸡、母鸡、小鸡各多少只？请用 JavaScript 语言设计程序求解。

6. 使用 JavaScript 语言编写程序，输出 1～10 的平方和，要求分别使用 3 种不同的循环语句实现：

（1）使用 for 语句实现。

（2）使用 while 语句实现。

（3）使用 do…while 语句实现。

7. 请使用 jQuery 编写一个图片动态隐藏动画效果的 Web 页面。

第5章 Photoshop基础

如果一个网站添加了精心设计的图片作点缀，会使网站看上去更加美观漂亮，吸引用户的眼球。因此，一个好的网页设计人员除了掌握前几章必须的基础知识外，还需要掌握网页图像的基本处理。

用于网页图像处理的软件有很多，但是使用最多、功能最强大的是 Photoshop，简称为 PS。Photoshop 是一款由 Adobe 公司开发的平面图像处理软件，主要用于平面设计、广告摄影、影像创意、网页制作、后期修饰、视觉创意、界面设计等领域。Adobe 公司于 2003 年 10 月发布了 Photoshop 8 并更名为 CS，此后从 2003 到 2012 年期间正式发行的 Photoshop 软件均以 CS 作为其版本编号前缀。2013 年 7 月 Adobe 公司发布的最新版本 Photoshop CC(Creative Cloud)则是采用了目前最流行的云服务技术。本书所有实例均以 2007 年 4 月发行的 Photoshop CS3 为版本详细讲解图像处理基本方法。

由于 Photoshop 主要处理的是由像素构成的数字及图像，所以 Photoshop 支持的图像格式大多是位图格式，例如 JPEG、TIFF、GIF、BMP、PNG 等。其中，JPEG、GIF 和 PNG 图像格式在网页设计中会经常用到，JPEG 和 GIF 图像在存储时经过压缩处理，文件通常比较小，而 PNG 图像格式则是 Fireworks 图像处理软件专用的一种图像格式。除此之外，Photoshop 还有自己专用的图像格式 PSD。PSD 格式最大的优点就是可以存储图像处理的所有细节信息，例如图层、通道、蒙版和色彩信息，最大程度保留了图像原始信息，非常有利于对未完成或需要反复修改的图像进行保存，缺点也是显而易见的，就是存储容量大。

5.1 Photoshop 工作环境

双击 Ps 图标，启动 Photoshop CS3，进入 Photoshop CS3 工作界面，如图 5.1 所示。

Photoshop CS3 工作界面由以下几部分组成：

- 菜单提供了 Photoshop 图像处理的全部功能。
- 属性栏又称为工具选项栏，根据所选的工具进行相应的参数设置。
- 工具箱提供了图像处理的基本工具，主要有图像选择工具、图像绘制/修饰工具、文字编辑工具、色彩填充工具以及辅助工具等。
- 图像窗口也称为画布窗口，主要用来显示图像文件。Photoshop 支持同时打开多个图像文件，可以利用图像窗口的最大、最小和关闭按钮对其进行管理。
- 浮动面板提供图像处理相关的辅助功能，主要有导航器面板、直方图面板、颜色样式



图 5.1　Photoshop CS3 工作界面

面板、历史记录面板、图层面板等，所有浮动面板都可以通过单击“窗口”菜单中的菜单项打开或关闭。其中，导航器面板主要用于对整个图像的导航浏览，可以通过调整面板下方的活动滑块改变图像窗口中的图像显示比例，面板中的红色显示框始终定位到当前图像窗口中显示的图像。当显示的图像大于图像窗口时可用鼠标拖动面板内红色显示框改变图像在窗口中的显示区域。历史记录面板可以记录图像处理过程中的每一步操作，存储的操作是有限的。设计人员可以在历史记录面板通过拖动滑块轻松进行撤销操作。导航器面板和历史记录面板如图 5.2 和图 5.3 所示。



图 5.2　导航器面板



图 5.3　历史记录面板

5.2　Photoshop 基本工具

Photoshop CS3 中的工具箱提供了图像处理的各种基本工具，按照功能可分为：图像选区工具、图像绘画和修饰工具以及其他辅助工具。

5.2.1　图像选区

使用 Photoshop 对图像进行处理的时候大多数并不是针对整幅图像，这就需要使用工具箱中的选区工具将图像中需要处理的区域创建为选区，然后再执行其他操作。

1. 选区创建

Photoshop CS3 提供了多种选区工具以方便快速创建不同形状的选区，选区一旦创建成功其边界会以虚线框显示。如果需要取消当前选区，可以同时按住键盘上的 Ctrl+D 键。

(1) 选框工具组

选框工具组是创建图像选区的最基本的工具，可以用来创建长方形、正方形、椭圆和正圆等规则形状选区，它包括一组选区工具：矩形选框工具 、椭圆选框工具 、单行选框工具 和单列选框工具 。

将鼠标移到工具箱中矩形选框工具图标 ，单击鼠标左键按住 1 秒钟，显示弹出菜单，单击对应图标即可选中该工具，如图 5.4 所示。

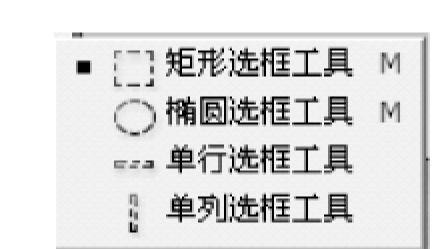


图 5.4　选框工具组工具

- 矩形或椭圆选框工具

使用矩形或椭圆选框工具可以创建外形为矩形或椭圆形状的选区，具体操作过程如下：

① 在工具箱中选择矩形选框工具 或椭圆选框工具 。

② 在图像窗口中需要创建为选区的图像区域中单击鼠标左键并拖动鼠标即可绘制出矩形或椭圆选区，选区的边界会以闪动的虚线框表示，如图 5.5 所示。

如果想创建正方形或圆形的选区，需要在创建选区单击鼠标左键并在拖动的同时按住 Shift 键。





图 5.5　创建矩形或椭圆选区

- 单行或单列选框工具

使用单行或单列选框工具可以创建只有一个像素高的行(单行)或一个像素宽的列(单列)的形状的选区，具体操作过程如下：

① 在工具箱中选择单行选框工具 或单列选框工具 。

② 在图像窗口中单击鼠标左键，在单击的位置自动创建单行或单列选区，如图 5.6 所示。

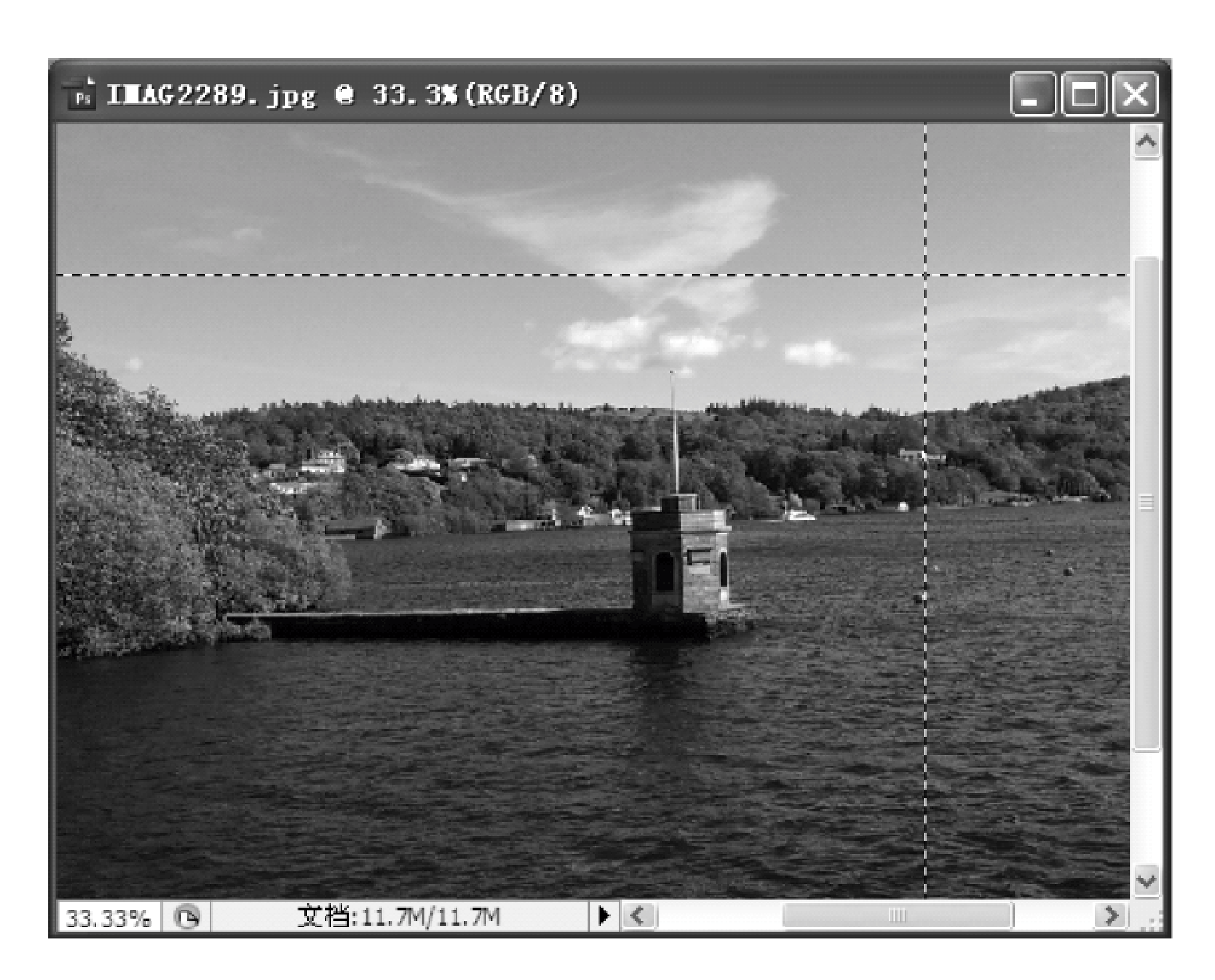


图 5.6 创建单行或单列选区

(2) 套索工具组

套索工具组常用来创建外形为任意不规则形状的选区，它包括一组工具：套索工具、多边形套索工具和磁性套索工具。

将鼠标移到工具箱中套索工具图标，单击鼠标左键按住1秒钟，显示弹出菜单，单击对应图标即可选中该工具，如图5.7所示。

套索工具 L
多边形套索工具 L
磁性套索工具 L

图 5.7 套索工具组工具

提示：一旦选中相应的套索工具，鼠标就会自动变成该工具图标的形状，进入选区创建状态，如果想退出，只需要双击鼠标左键，套索工具会自动创建当前鼠标位置到起点的闭合选区，此时按住键盘上的Ctrl+D键取消选区，即可重新选取。

• 套索工具

如果要选取的图像轮廓为任意不规则形状，可使用套索工具创建该选区，具体操作过程如下：

① 在工具箱中选择套索工具。

② 在图像窗口中单击鼠标左键沿着要选取图像轮廓边缘拖动鼠标，此时鼠标变成形状。

③ 当拖动鼠标回到起点位置时，松开鼠标左键，即可建立一个该图像轮廓形状的不规则选区，如图5.8所示。

• 多边形套索工具

如果要选取的图像边缘多由直线连接的不规则形状，可使用多边形套索工具创建该选区，具体操作过程如下：

① 在工具箱中选择多边形套索工具。

② 在图像窗口中沿着要选取的图像轮廓单击鼠标左键作为起始点，此时鼠标变成形状。然后移动鼠标到需要改变方向的位置单击鼠标，插入锚点，Photoshop会用直线将这



图 5.8　使用套索工具创建选区

些插入的锚点连接起来。

③ 当移动鼠标回到起点位置时，鼠标右下角出现小圆圈，单击鼠标左键，此时自动建立封闭的不规则多边形选区，如图 5.9 所示。



图 5.9　使用多边形套索工具创建选区

- 磁性形套索工具

如果要选取的图像边缘与周围图像的颜色对比鲜明，可以使用磁性套索工具快速、准确地选取。具体操作过程如下：

① 在工具箱中选择磁性套索工具。

② 在图像窗口中沿着要选取的图像边缘移动鼠标，此时鼠标自动变成形状。磁性

套索工具会自动对边界进行分析插入锚点，此时不需要按住鼠标按键。

③ 当移动鼠标回到起点时，鼠标右下角出现一个小圆圈，单击鼠标左键即可创建选区，如图 5.10 所示。



图 5.10　使用磁性套索工具创建选区

在使用磁性套索工具时，可以通过设置该工具属性栏的"对比度"和"频率"参数提高选取的准确度，如图 5.11 所示。



图 5.11　"对比度"和"频率"参数设置

对比度：取值范围是 1%～100%，用来设置磁性套索工具边缘图像检测的灵敏度。如果要选取的图像边界颜色对比较强烈，需要设置较高的百分数值。反之，则输入较低的百分数值，默认值为 10%。

频率：取值范围是 0～100，用来设置添加到路径中锚点的密度。数值越大，进行边界分析时自动插入的锚点就越多，选取的图像边缘准确度就越高。因此，如果要选取的图像边缘比较复杂，就需要设置较高的数值，增加较多的锚点来提高选取图像边缘的准确性。默认值为 57。

(3) 魔棒工具组

为了方便快速创建不规则选区，Photoshop 还提供了魔棒工具组工具。它包括两个工具：快速选择工具和魔棒工具。

将鼠标移到工具箱中魔棒工具图标，单击鼠标左键按住 1 秒钟，显示弹出菜单，单击对应图标即可选中该工具，如图 5.12 所示。

快速选择工具　W
魔棒工具　W

图 5.12　魔棒工具组工具

- 快速选择工具

快速选择工具是从 Photoshop CS3 版本开始增加的一个工具，它可以通过调整画笔的直径、硬度和间距等参数而快速控制选择区域的大小。

一般来说，画笔直径设置越大，选择区域就大，选取的效率就高，但选择粗糙，容易多选；

相反，小画笔选择慢，但得到的边缘精确度较高。画笔的直径参数是通过快速选择工具的属性栏中"画笔"进行设置，如图 5.13 所示。

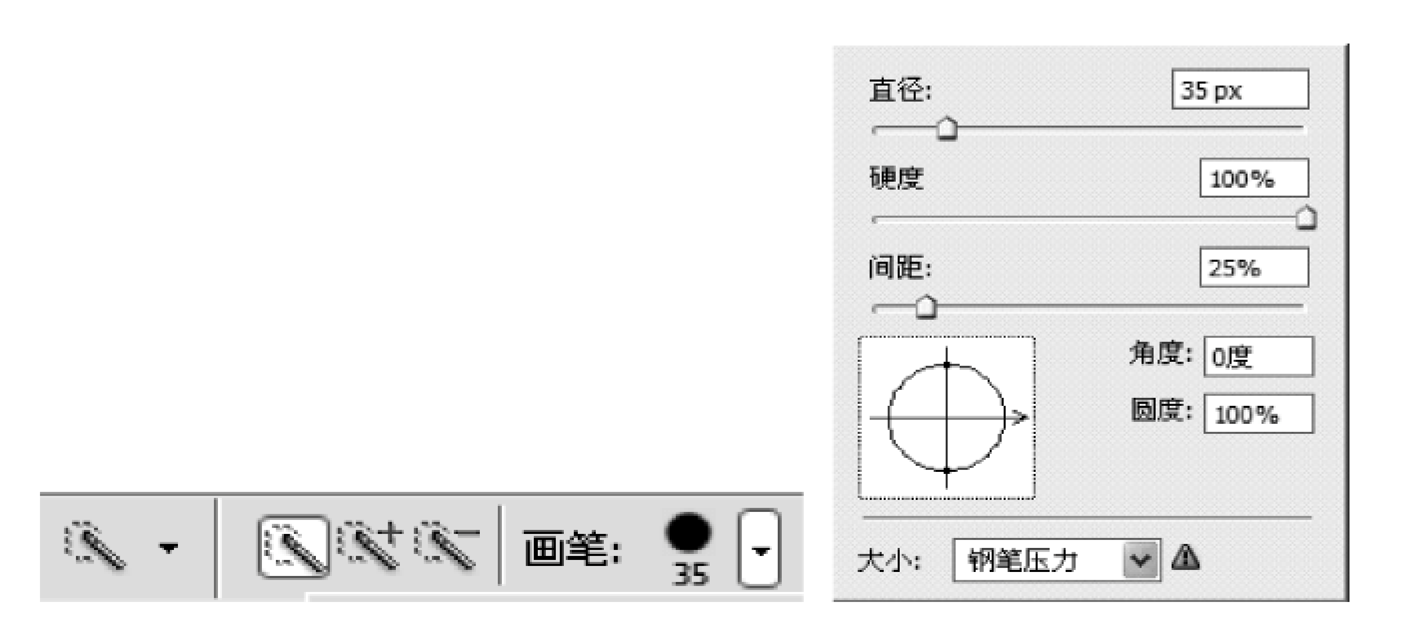


图 5.13 快速选择工具"画笔"参数设置

在快速选择工具的属性栏中还可以设置选区方式。设置选区方式有利于调整选区大小，使创建的选区更精确。

新建选区：单击 图标，每次单击鼠标时都会自动创建新选区。

增加选区：单击 图标，每次单击鼠标时都会将当前区域自动添加到已有选区中。

减去选区：单击 图标，每次单击鼠标时都会从已有选区中减去当前区域。

在使用快速选择工具创建选区时，鼠标会变成"＋"字形状，此时只要单击同时拖动鼠标，所建选区会向外自动扩展并查找和跟随图像中定义的边缘，如图 5.14 所示。



图 5.14 使用快速选择工具创建选区

• 魔棒工具

如果要选取的图像区域颜色比较相近，可以使用魔棒工具。魔棒工具是基于图像中相邻像素的颜色近似程度来进行选择的。

魔棒工具的属性栏中"容差"和"连续"参数对设置选区很有帮助，如图 5.15 所示。

容差：用来设置选取区域的颜色近似程度。取值范围是 0～255，默认值为 32。值越小，选取的区域中颜色越接近，值越大，选取的区域中颜色越多，选取范围就越大。

连续：用来设置要选取的区域是否连接起来形成一片。如果取消选中该选项，则会将当前图像中所有颜色相近的区域设置为选区。

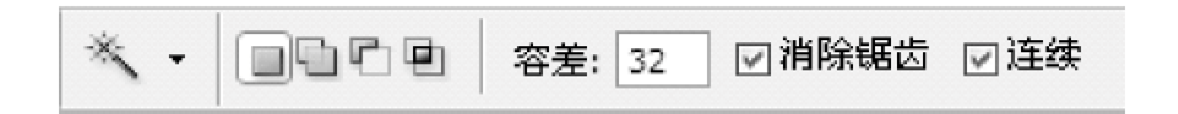


图 5.15　魔棒工具“容差”和“连续”参数设置

在使用魔棒工具创建选区时，鼠标会变成 形状，设置好“容差”参数，在需要创建为选区的位置单击鼠标即可，如图 5.16 所示。

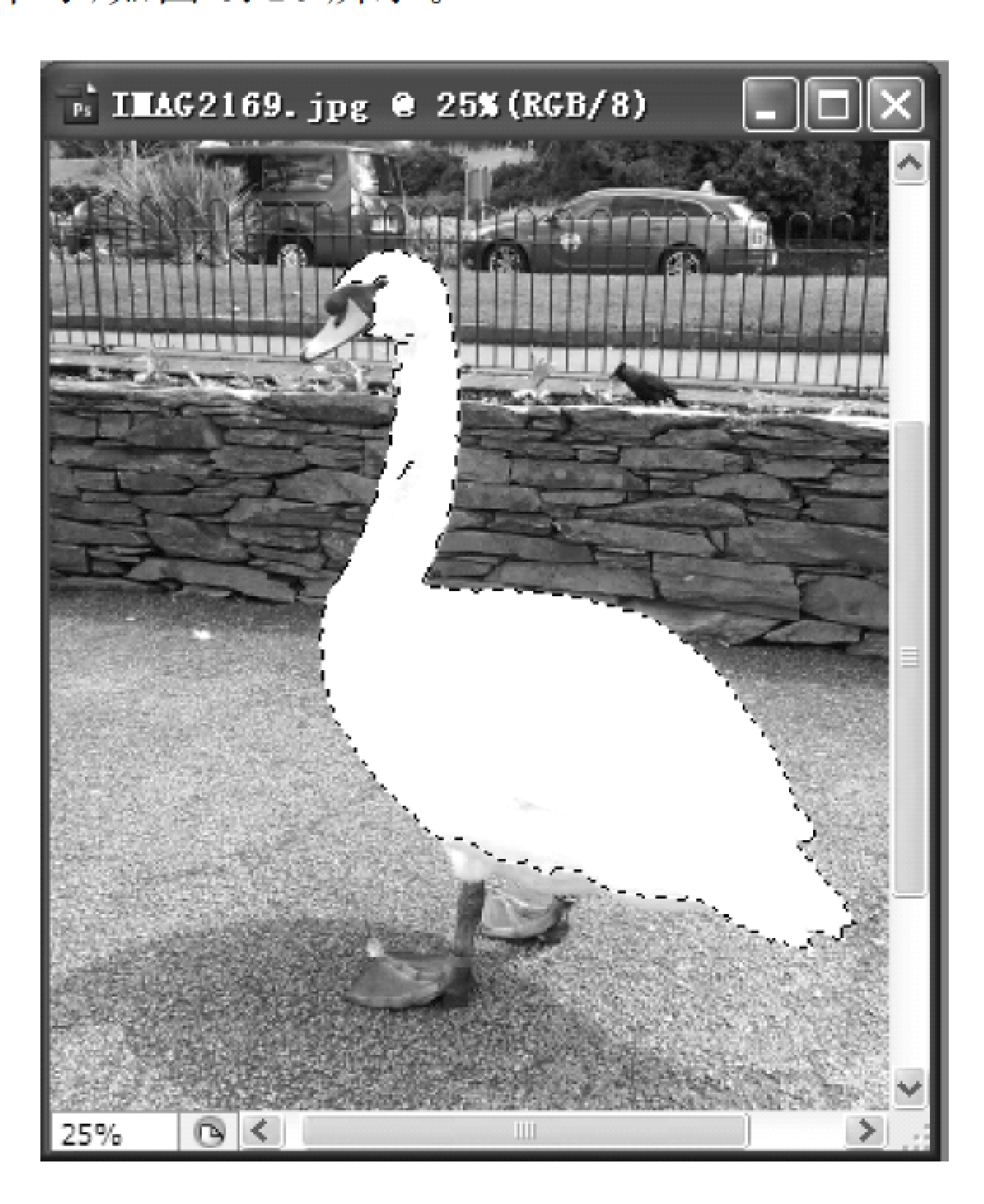


图 5.16　使用魔棒工具创建选区

提示：魔棒工具是通过容差值来调节选择区域，一次只能选择“一片”区域。如果想选择整个图像中相似的颜色区域，可以先用魔棒来选择某一块颜色，然后在菜单里选择“选择”→“选取相似”即可。

2. 选区操作

有时候要选取的图像比较复杂，不是一次就能得到所需的选区，需要进一步对初步选定的区域进行各种调整操作，以便符合最终需要。这就需要对选区进行操作。

选区的基本操作包括：移动选区、增减选区范围、消除锯齿和羽化选区四部分。

(1) 移动选区

移动选区只是移动选区的虚线边框。移动选区之前要确保已经选择了相应的选区工具，然后将鼠标移到选区内，鼠标变成 形状，此时单击鼠标左键并拖动即可移动选区。

(2) 增减选区范围

增减选区范围目的是要选取的图像区域更加符合需求，提高选取的精确度，主要包括四种选区操作： 新建选区、 添加选区、 选区相减和 选区相交，可以通过单击选区工

具的属性栏中对应图标完成设置。增减选区范围后的效果如图 5.17 所示。

新建选区：建立新的选区，这是选区工具的默认设置。

添加选区：将当前选区与原有选区进行合并，扩大选区范围。

选区相减：从原有选区中剔除当前选区，缩小选区范围。

选区相交：将当前选区和原有选区相交的区域保留下来形成新的选区。



(a) 添加选区的过程　(b) 添加选区后的效果　(c) 选区相减的过程

(d) 选区相减后的效果　(e) 选区相交的过程　(f) 选区相交后的效果

图 5.17　增减选区范围后的效果

提示：在使用选区工具增减选区范围时，应首先选中对应的选区工具，然后在该工具的属性栏中单击对应图标设置选区操作，接下来就可以进行图像选取操作了。对比较复杂的图像进行选取时，可能会联合使用多个选区工具，并反复对所选区域进行增减选区范围操作。

(3) 消除锯齿

在使用选区工具时，属性栏中一般都会出现消除锯齿选项。该选项用于消除选区边缘的锯齿，使创建的选区边缘变得比较平滑。默认情况下，该选项是处于选中状态。

(4) 羽化选区

在进行图像合成时，为了让创建的选区边缘变得比较柔和，使选区内的图像能很好地与外部其他图像自然过渡，达到自然融合的效果，需要对选区进行羽化操作。图像边缘过渡的宽度称为羽化半径，单位为像素。图像羽化后的效果如图 5.18 所示。

在 Photoshop 中，羽化选区操作可分为两种情况：①在创建选区前设置羽化半径，这需要在选区工具的属性栏中对羽化半径进行设置，默认情况下，其值为 0；②在创建选区后再进行羽化，这需要在当前创建的选区中单击鼠标右键，弹出的菜单中选择“羽化”，然后在“羽化选区”对话框中设置羽化半径，如图 5.19 所示。



图 5.18　羽化效果



图 5.19　羽化半径设置

5.2.2　图像绘画与修饰

Photoshop 工具箱中除了前一小节介绍的选区工具外，还有用于图像绘画与修饰的工具，主要包括画笔工具组、历史画笔工具、文字工具、填充工具、图章工具、擦除工具以及图像修复和修饰工具，下面分别介绍。

1. 画笔工具组

画笔工具组主要用来辅助绘图，精确绘图需要使用钢笔工具。Photoshop 预设了多种笔刷效果，可以使用画笔工具绘制出各种效果，它包括一组工具：画笔工具 、铅笔工具 和颜色替换工具 。

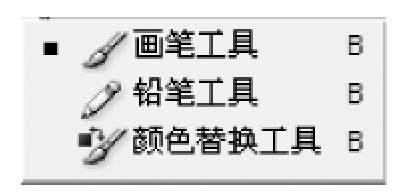


图 5.20　画笔工具组工具

将鼠标移到工具箱中画笔工具图标 ，单击鼠标左键按住 1 秒钟，显示弹出菜单，单击对应图标即可选中该工具，如图 5.20 所示。

(1) 画笔工具

画笔工具绘制出的线条更加柔和，其效果如同毛笔绘制的线条。使用画笔工具需要在属性栏中设置“不透明度”和“流量”参数。要想设置画笔的大小需要在属性栏中单击“画笔”右侧的下拉箭头，在弹出的面板中设置画笔主直径、硬度以及笔刷。画笔工具的参数设置如图 5.21 所示。

不透明度：用来设置绘制线条的不透明效果。

流量：用来设置绘制线条的浓淡效果。

主直径：用来设置画笔的大小，单位为像素。

硬度：用来设置画笔的边缘柔和程度。其值为百分数，百分比越低，画笔绘制出的线条边缘越模糊。反之，则绘制出的线条边缘越清晰。

笔刷：Photoshop 自带有多种笔刷效果，可以绘制各种效果的线条。

使用画笔工具绘制图像时，应首先在工具箱中选中该工具，然后在属性栏中设定好相关

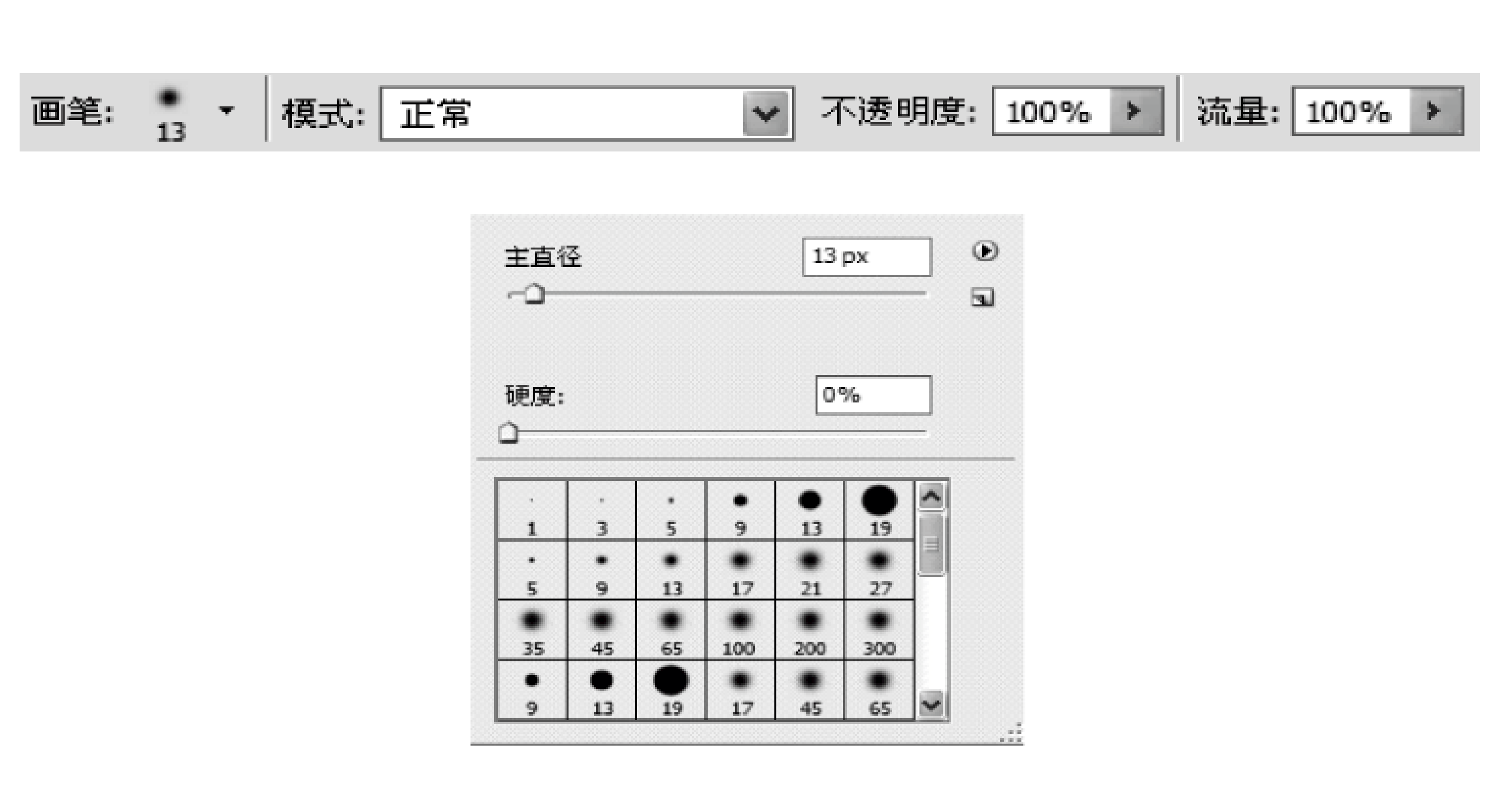


图 5.21 画笔工具的参数设置

参数,在图像窗口中单击鼠标左键移动鼠标。在单次操作中,即单击鼠标左键绘制不松开,流量具有颜色叠加属性,而不透明度没有。使用画笔工具绘制线条效果如图 5.22 所示。

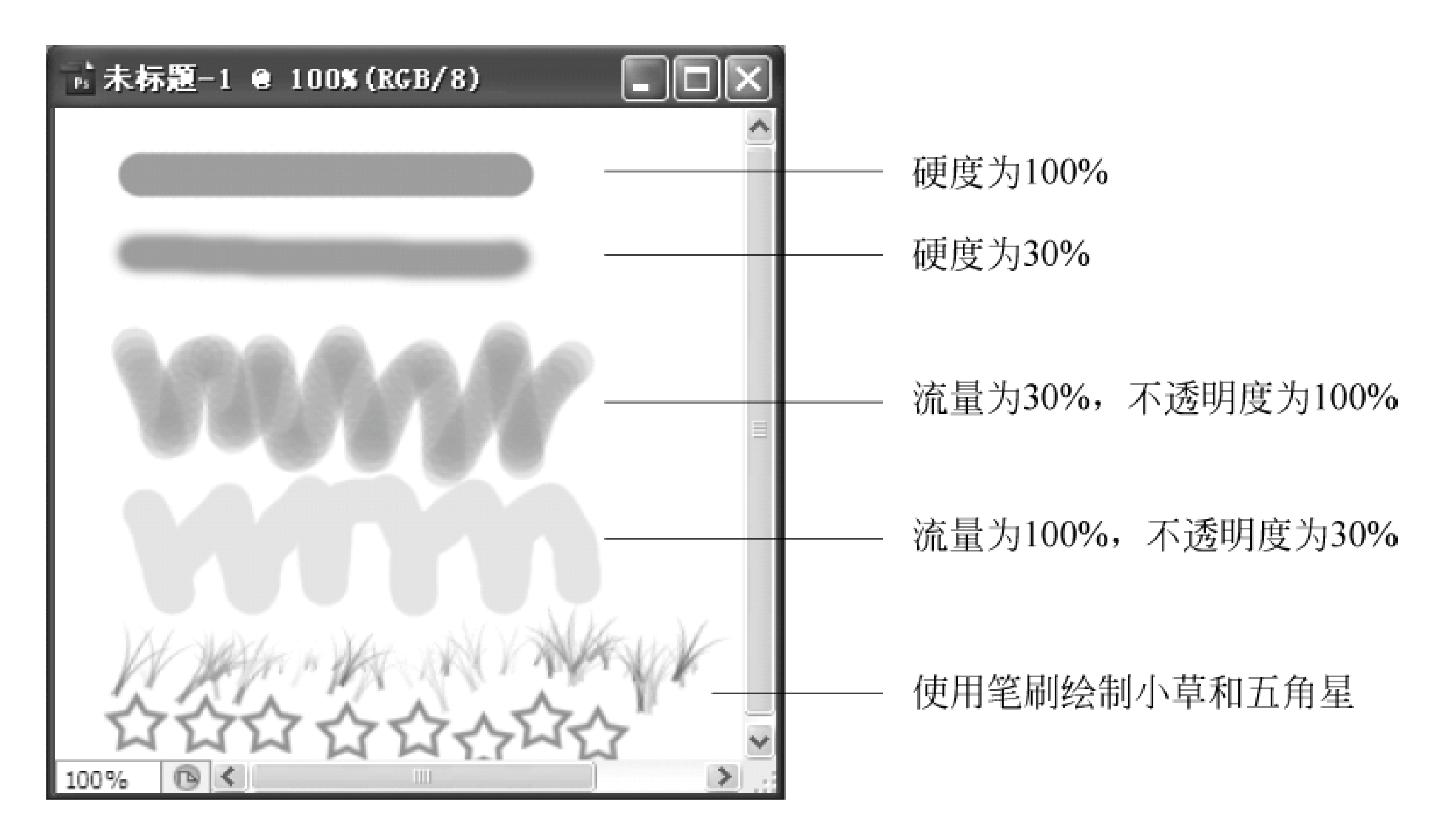


图 5.22 使用画笔工具绘制线条

除了自带的笔刷效果外,如果想要使用自定义的图案作为笔刷,需要首先定义画笔预设。方法如下:

① 使用任意选区工具选中需要定义为笔刷的图案区域。

② 选择菜单“编辑”→“定义画笔预设”,在弹出的“画笔名称”对话框中自定义笔刷名称,单击“确定”按钮。Photoshop 会将选区中的图案作为自定义的笔刷效果添加到画笔工具中。然后再使用画笔工具时,就可以在弹出面板中选择该笔刷。使用自定义笔刷绘制效果如图 5.23 所示。

(2) 铅笔工具

与画笔工具不同,铅笔工具常用来绘制比较硬朗的线条,其使用方法和画笔工具类似,在属性栏中设置好铅笔的直径和硬度参数就可以绘制,如图 5.24 所示。

(3) 颜色替换工具

简单来说,颜色替换工具就是将图像中的某些特定颜色替换成其他颜色。具体操作过程如下:

① 在图像窗口中,使用选区工具将需要替换的颜色区域创建为选区。




(a) 创建选区



(b) 使用自定义笔刷绘制

图 5.23　使用自定义笔刷绘制

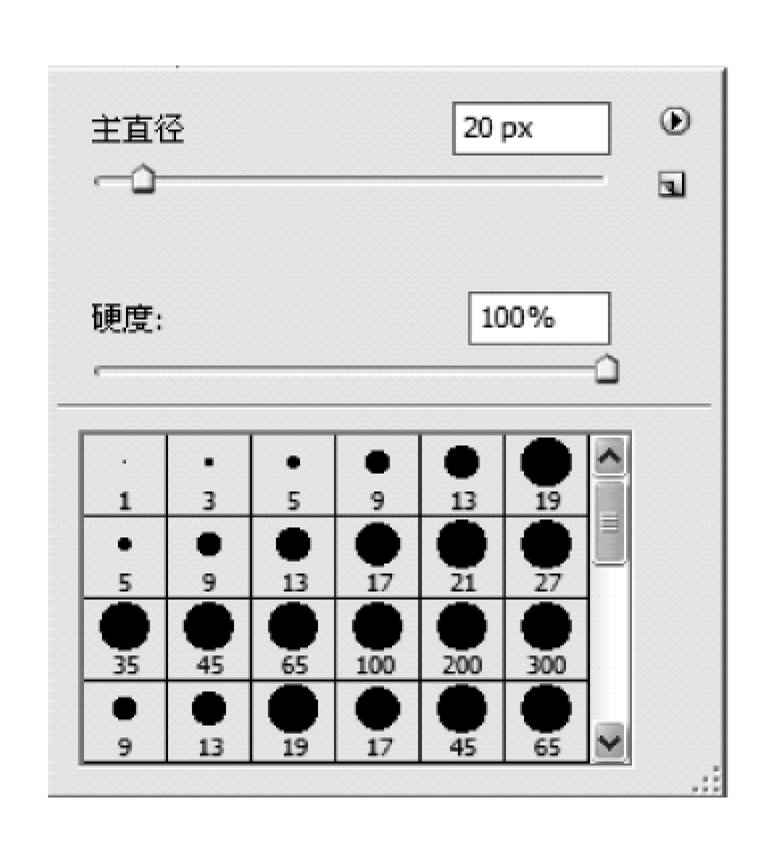
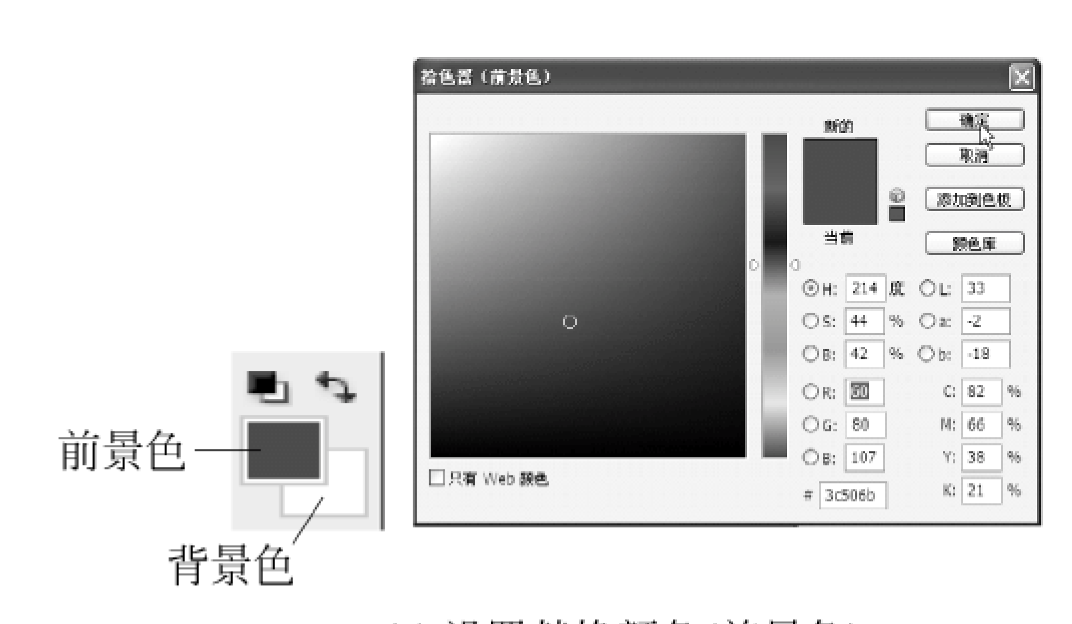


图 5.24　铅笔工具的参数设置

② 在工具箱中，单击前景色/背景色图标，在弹出的"前景色"对话框中设置需要替换的颜色(即前景色)。

③ 在工具箱中选择颜色替换工具。然后在选定的区域内，单击鼠标左键进行移动，选区内的颜色即被替换成前景色。使用颜色替换工具替换图像颜色效果如图 5.25 所示。



(a) 设置替换颜色(前景色)



(b) 替换选区内的颜色

图 5.25　使用颜色替换工具替换图像颜色

如果想要得到更精确的颜色替换，则需要在颜色替换工具属性栏中设置颜色替换时采用的取样模式和容差值。关于取样模式，在后面的橡皮擦工具中会进一步描述。

2. 历史画笔工具

Photoshop 提供的历史画笔工具不同于一般绘图工具，需要借助历史记录面板完成图

像恢复操作。历史画笔工具包括：历史记录画笔工具 和历史记录艺术画笔工具 。

将鼠标移到工具箱中历史记录画笔工具图标 ，单击鼠标左键按住 1 秒钟，显示弹出菜单，单击对应图标即可选中该工具，如图 5.26 所示。

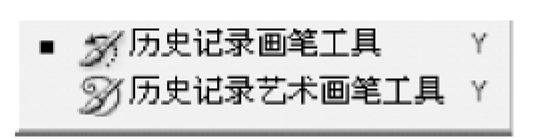


图 5.26 历史画笔工具

(1) 历史记录画笔工具

历史记录画笔工具主要用来恢复图像。在 Photoshop 中，对图像的每一步操作都记录到历史记录面板中。首先在历史记录面板中某一操作前单击鼠标左键设置还原点，然后在图像窗口中单击鼠标并移动，此时历史记录画笔工具就会将图像恢复到还原点设置时的操作状态下。使用历史记录画笔工具恢复图像效果如图 5.27 所示。



(a) 设置还原点

(b) 恢复之前



(c) 恢复之后

图 5.27 使用历史记录画笔工具恢复图像

(2) 历史记录艺术画笔工具

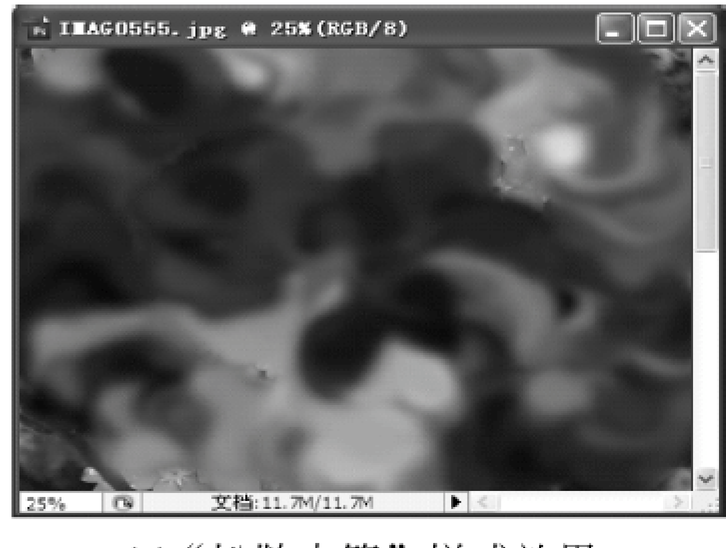
历史记录艺术画笔工具使用方法与历史记录画笔工具类似，不同之处在于，历史记录艺术画笔工具在恢复图像的同时，对图像添加了艺术效果。历史记录艺术画笔工具提供了十种样式，在属性栏中的“样式”下拉列表框中选择所需要的样式。使用历史记录艺术画笔工具处理图像效果如图 5.28 所示。



(a) 样式设置



(b) “绷紧短” 样式效果



(c) “松散中等” 样式效果

图 5.28 使用历史记录艺术画笔工具处理图像

3. 文字工具

Photoshop 提供的文字工具是一组工具，包括横排文字工具 T 、直排文字工具 IT 、横排文字蒙版工具 和直排文字蒙版工具 。

将鼠标移到工具箱中横排文字工具图标 T ，单击鼠标左键按住 1 秒钟，显示弹出菜单，单击对应图标即可选中该工具，如图 5.29 所示。

(1) 横排文字工具和直排文字工具

横排文字工具和直排文字工具主要用来为图像添加横向排列和纵向排列的文字，Photoshop使用文字图层存储文字信息。

在使用该工具时需要在属性栏中设定文字的字体、大小、颜色以及排列方式等。接下来在图像窗口中单击鼠标左键，出现闪动光标提示文字输入，如图5.30所示。



图5.29　文字工具　　　　图5.30　使用横排文字工具输入文字

• 更改文字属性设置

如果想要对已经输入的文字更改颜色、大小等参数设置，则需要选择对应文字工具，然后在图层面板选中该文字所在图层，此时鼠标变成 I 形状，然后单击并拖动鼠标，将文字选中，在属性栏中更改对应参数设置即可，如图5.31所示。单击工具箱中其他工具图标，即可退出文字编辑状态。



(a) 选择文字图层



(b) 选择文字

图5.31　选择文字

• 文字变形

如果想要对文字实现更复杂的变形操作，例如旋转、缩放等，则需要使用自由变换功能。具体操作过程如下：

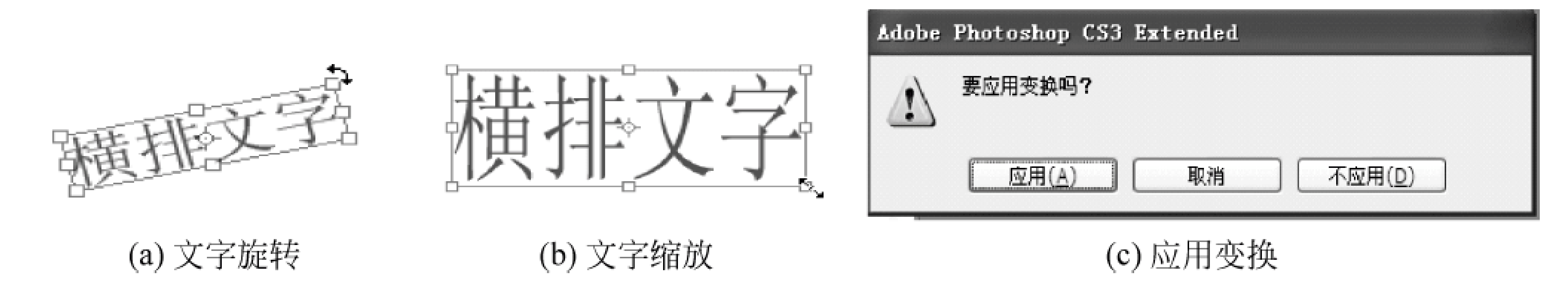
① 首先选中文字所在图层，然后单击菜单“编辑”→“自由变换”，文字四周会出现带锚点的矩形控制框。

② 旋转操作：将鼠标移到矩形控制框的任意一个角上，鼠标变成 ↻ 形状，单击鼠标左键按逆时针或者顺时针移动鼠标，即可实现文字的旋转操作。

③ 缩放操作：将鼠标移到矩形控制框的任意一个角上或者任意一条边上的中间锚点，鼠标变成 ↔ 形状，单击鼠标左键并向外或向内拖动鼠标，即可实现文字的缩放操作。

④ 当使用自由变换完成文字的变形操作后，可以单击工具箱中任意工具图标，就可以

退出自由变换的编辑状态。此时,Photoshop 会弹出对话框,确认是否应用变换,单击“确定”按钮即可,如图 5.32 所示。



(a) 文字旋转　　(b) 文字缩放　　(c) 应用变换

图 5.32　使用自由变换进行文字变形

(2) 横排文字蒙版工具和直排文字蒙版工具

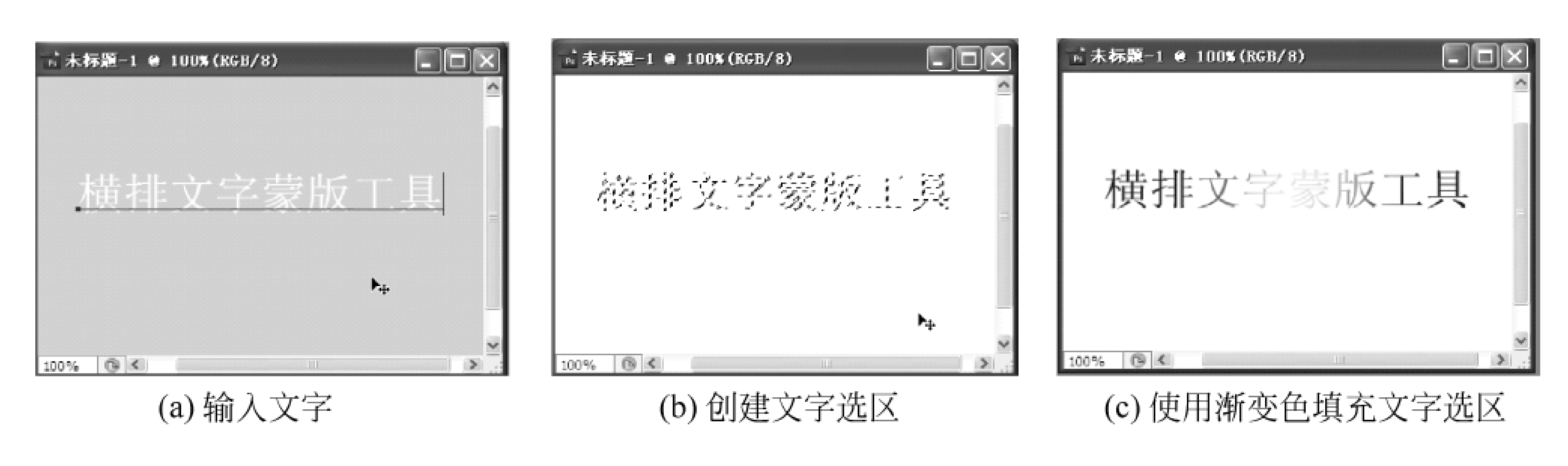
与横排文字和直排文字工具完全不同,横排文字蒙版工具和直排文字蒙版工具主要用来创建横向排列和纵向排列的文字形状的选区,使用该工具和颜色填充工具可以创建更加丰富多彩的文字效果。横排文字蒙版工具具体操作过程如下:

① 首先从工具箱中选择横排文字蒙版工具 。

② 在图像窗口中合适位置单击鼠标左键,出现闪动光标,在属性栏设置好横排文字蒙版工具的字体、大小以及排列方式等参数,然后在闪动光标处输入文字。此时,图像窗口除文字外背景颜色变成了淡红色。

③ 在工具箱中单击任意图标,退出文字蒙版编辑状态。输入的文字边界出现闪动的虚线框,此时文字选区创建成功。

文字选区一旦创建成功,接下来就可以使用颜色填充工具为该选区填充颜色、图案或渐变色。使用横排文字蒙版工具创建文字选区如图 5.33 所示。



(a) 输入文字　　(b) 创建文字选区　　(c) 使用渐变色填充文字选区

图 5.33　使用横排文字蒙版工具创建文字选区

4. 填充工具

Photoshop 提供的填充工具可以使用指定的颜色或图案对选区内进行填充,它包括一组工具:渐变工具 和油漆桶工具 。

将鼠标移到工具箱中渐变工具图标 ,单击鼠标左键按住 1 秒钟,显示弹出菜单,单击对应图标即可选中该工具,如图 5.34 所示。

(1) 渐变工具

渐变工具可以用来对指定区域内进行两种以上颜色的渐变填充。具体操作过程如下:

① 使用选区工具在图像区域中创建选区。

② 在工具箱中选择渐变工具 ，鼠标变成“+”字形状。

③ 设置要填充的渐变图案。

Photoshop 预设了多种渐变图案效果，可以在渐变工具的属性栏中单击 图标的下拉箭头，在弹出面板中单击需要填充的渐变图案，如图 5.35 所示。

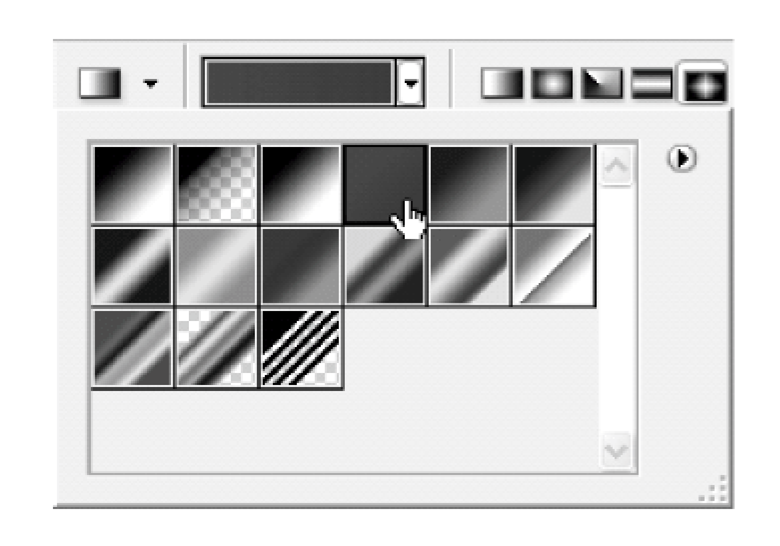


图 5.34 填充工具　　图 5.35 选择渐变图案

④ 设置渐变填充模式。

渐变工具提供了五种渐变填充模式：线性渐变 、径向渐变 、角度渐变 、对称渐变 以及菱形渐变 ，在属性栏中单击对应图标即可设置渐变填充模式。当设置不同的渐变填充模式时，其渐变填充方法和填充效果也不同，如图 5.36 所示。

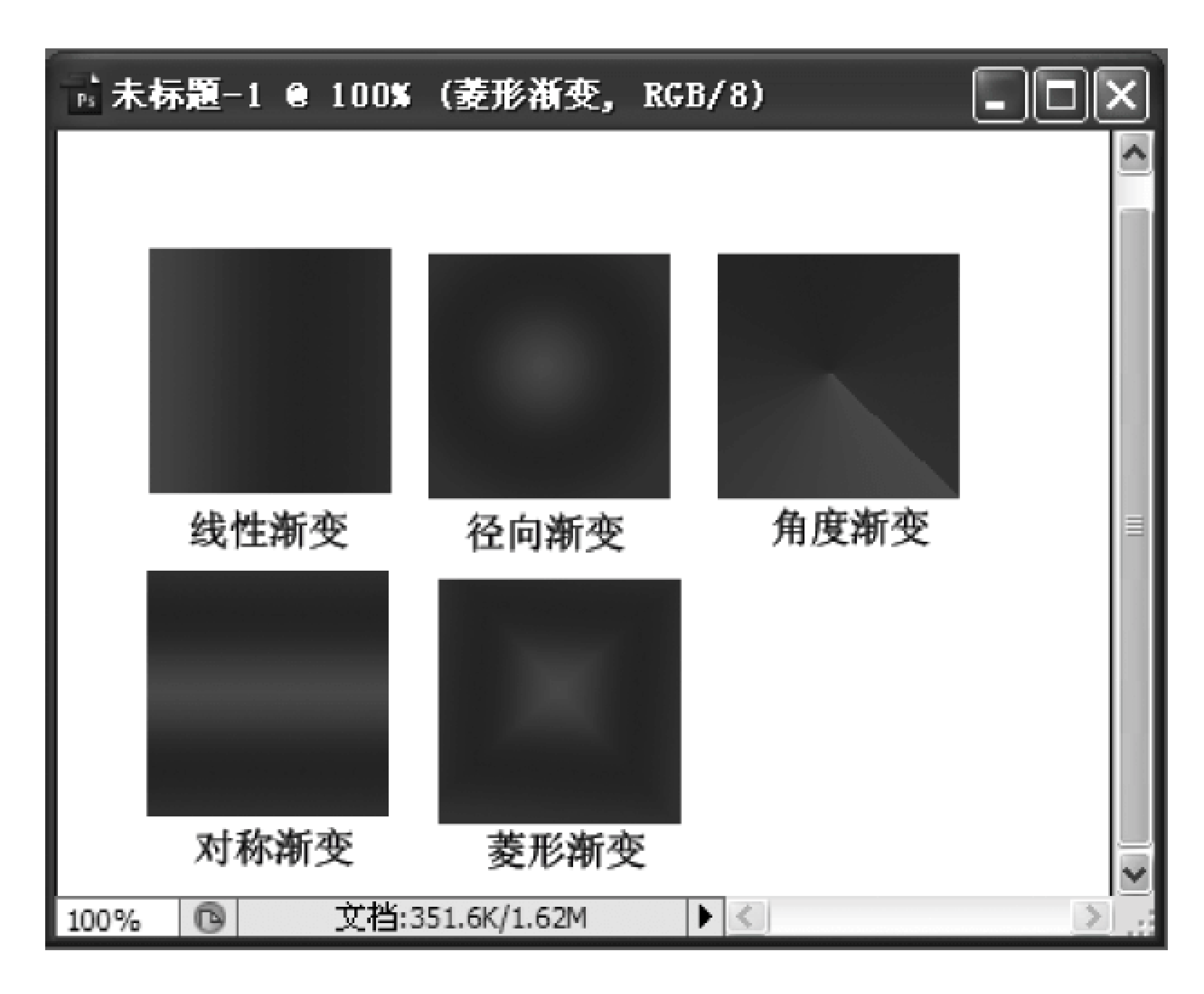


图 5.36 五种渐变模式

线性渐变：线性渐变是沿着直线方向从左(右)到右(左)或从上(下)到下(上)进行，渐变的起点位于选区边缘。

径向渐变：径向渐变是从中心向四周进行，渐变的起点位于选区中心。

角度渐变：角度渐变是沿着 360°角的方向旋转进行，渐变的起点到终点之间的连线作为渐变开始的 0°角方向。

对称渐变：对称渐变是从中心开始上下或左右对称渐变，渐变的起点位于选区中间。

菱形渐变：菱形渐变是从中心向四周呈菱形进行，渐变的起点位于选区中心。

⑤ 在设定的选区中进行渐变填充。

进行渐变填充时，单击鼠标左键，此时为渐变填充的起点位置，然后拖动鼠标直到松开

鼠标左键，此时为渐变填充的终点位置。在图像窗口中显示为从起点到终点之间连接的一条直线。一般情况下，颜色渐变则是从起点位置到终点位置进行，如图 5.37 所示。

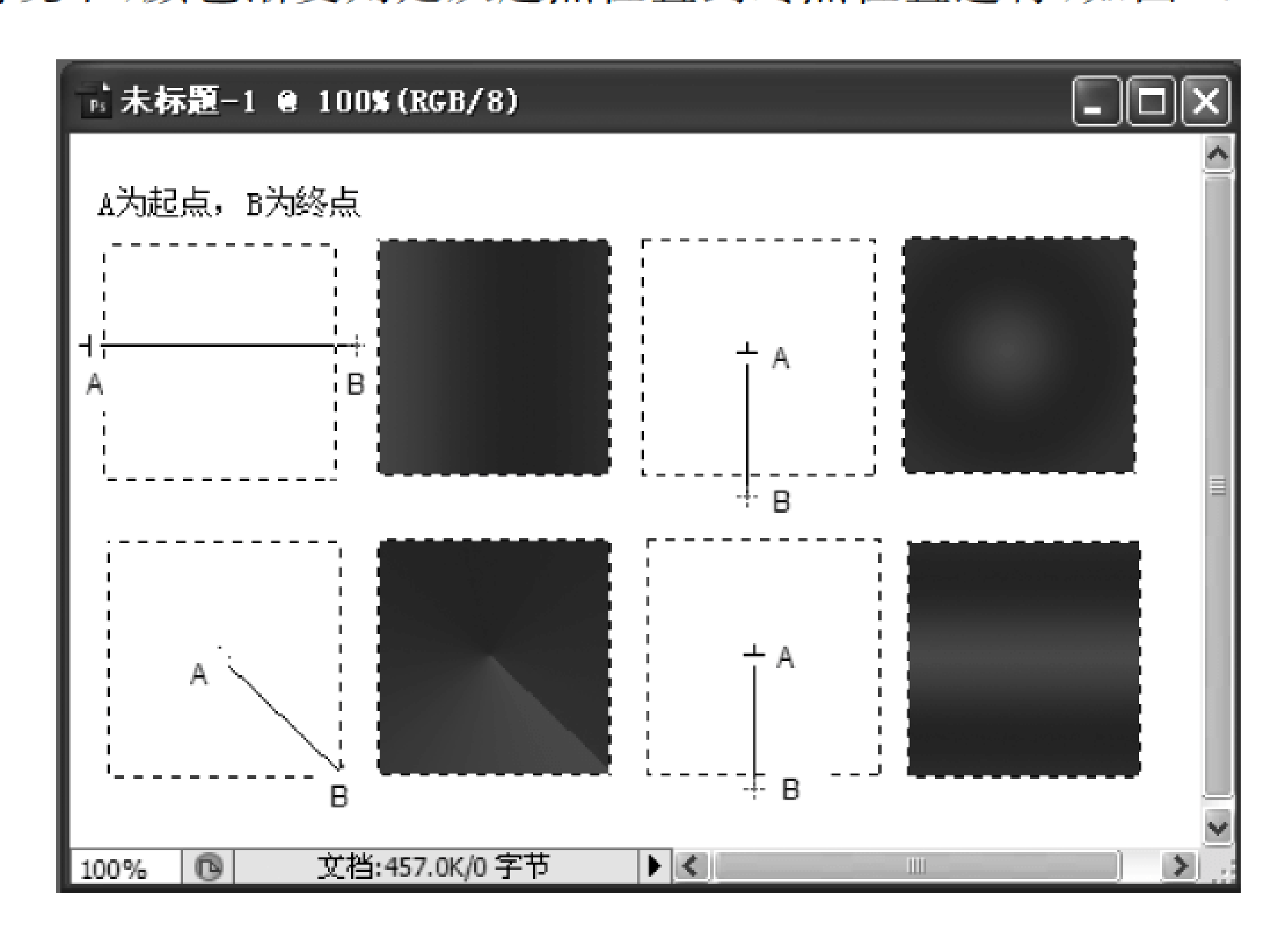
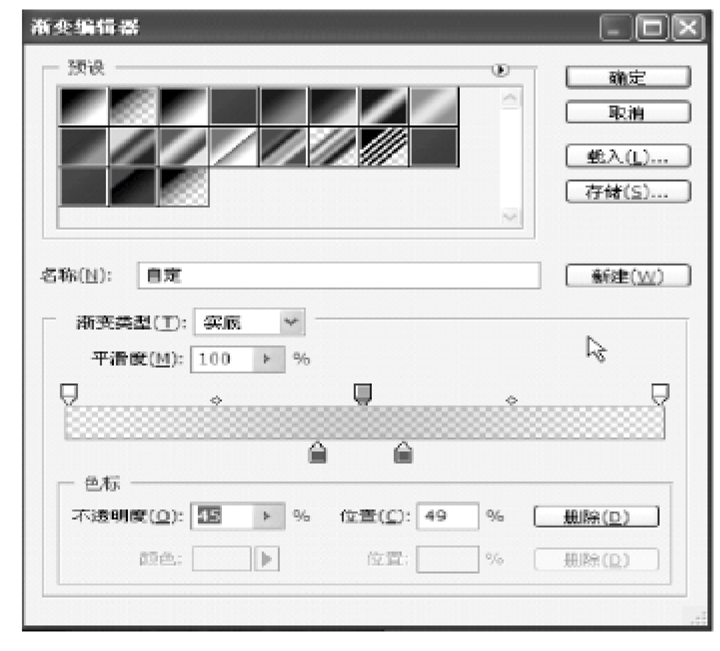


图 5.37　渐变工具填充渐变效果

除了使用预设的渐变图案，Photoshop 还允许设计人员自己自定义新的渐变图案。在属性栏中单击 图标的中间位置，显示“渐变编辑器”对话框，在“预设”选项组中选择一个与自定义的渐变图案类似的图案单击，并对其命名，然后单击“新建”按钮。“渐变编辑器”对话框下方会显示该渐变图案的颜色条，同时 Photoshop 会自动在“预设”选项组的末尾添加该渐变图案的副本。

单击颜色条的上方色标可以更改当前渐变颜色的透明度效果，单击下方色标可以更改当前渐变的颜色，如果在空白色单击鼠标，还会添加新的色标，用来设置新增加的渐变色的颜色和透明度，然后单击“确定”按钮，就可以使用自定义的渐变图案填充图像。有时候为了达到朦胧的效果，需要在属性栏中设置渐变工具的“不透明度”参数。自定义渐变图案如图 5.38 所示。



(a) 渐变编辑器



(b) 使用自定义渐变图案填充效果

图 5.38　自定义渐变图案

（2）油漆桶工具

油漆桶工具可以用来在图像中填充颜色或图案。在进行填充时，油漆桶工具会根据设定的容差值，只对与鼠标单击处的颜色近似的区域进行填充，填充的效果是成“片”状区域。

油漆桶有两种填充方式：前景填充和图案填充。使用前者填充时，需要预先设置好要填充的前景色，然后单击鼠标左键进行填充。使用后者填充时，需要选择要填充的图案。在属性栏中设置好填充方式以及填充的“不透明度”和“容差”，在图像窗口中单击鼠标，即完成填充。图5.39给出了油漆桶属性栏设置，图5.40给出了使用“前景”和“图案”填充图像效果。

油漆桶也支持自定义图案填充，首先使用矩形选区工具创建矩形选区，然后单击菜单“编辑”→“定义图案”，选定的图案自动存储到油漆桶的图案中。

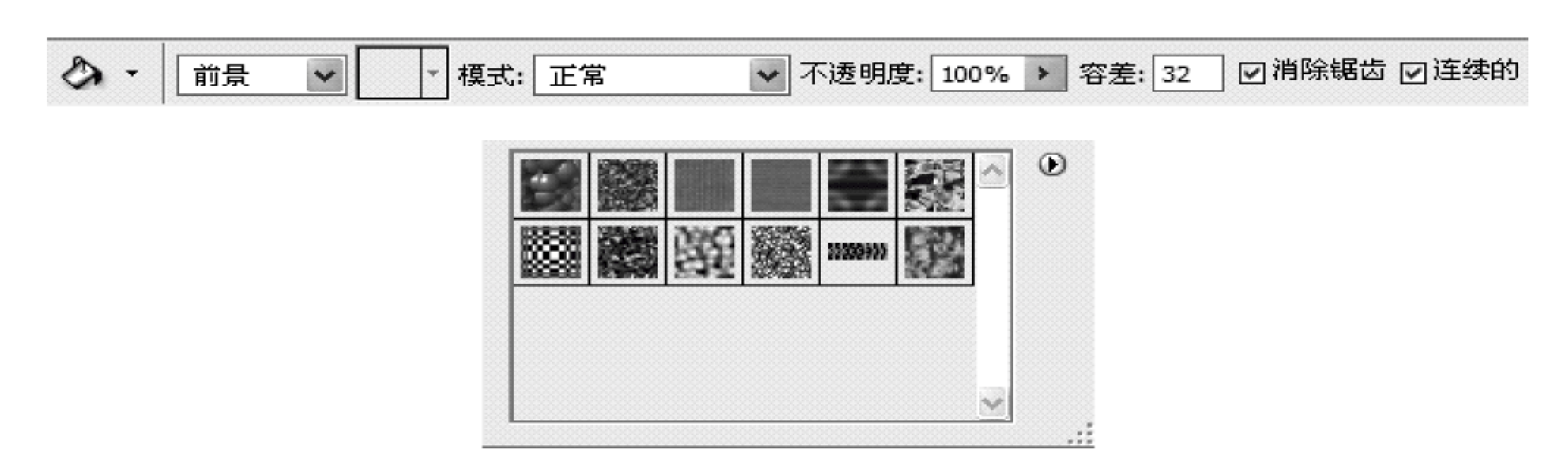


图5.39　油漆桶属性栏设置



(a) 使用“前景”填充



(b) 使用“图案”填充

图5.40　使用“前景”和“图案”填充图像

5. **图章工具**

图章工具主要用于图像的简单复制，它是一组工具包括：仿制图章工具 和图案图章工具 。

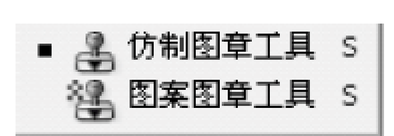


图5.41　图章工具

将鼠标移到工具箱中仿制图章工具图标 ，单击鼠标左键按住1秒钟，显示弹出菜单，单击对应图标即可选中该工具，如图5.41所示。

(1) 仿制图章工具

使用仿制图章工具复制图像时，需要首先在要复制图像的位置设置取样点，同时单击鼠标左键和Alt键，此时鼠标变成⊕形状。然后松开Alt键，将鼠标移到需要复制图像的目标区域，单击鼠标左键并移动，仿制图章工具会将“＋”光标位置的图像复制到鼠标移动的目标位置。使用仿制图章工具复制图像如图5.42所示。

(2) 图案图章工具

图案图章工具以预先设定好的图案为复制对象进行复制，具体操作过程如下：

① 将要复制图案的目标区域设定为选区。

② 在工具箱中选择图案图章工具 ，在属性栏中选择要复制的图案，然后单击鼠标左键并移动，此时，会将选择的图案复制到鼠标移动位置。使用图案图章工具复制图像如图 5.43 所示。

Photoshop 支持自定义图案，方法与自定义填充图案一致。



图 5.42　使用仿制图章工具复制图像

6. 擦除工具

Photoshop 提供的擦除工具有三个：橡皮擦工具 、背景橡皮擦工具 和魔术橡皮擦工具 。

将鼠标移到工具箱中橡皮擦工具图标 ，单击鼠标左键按住 1 秒钟，显示弹出菜单，单击对应图标即可选中该工具，如图 5.44 所示。

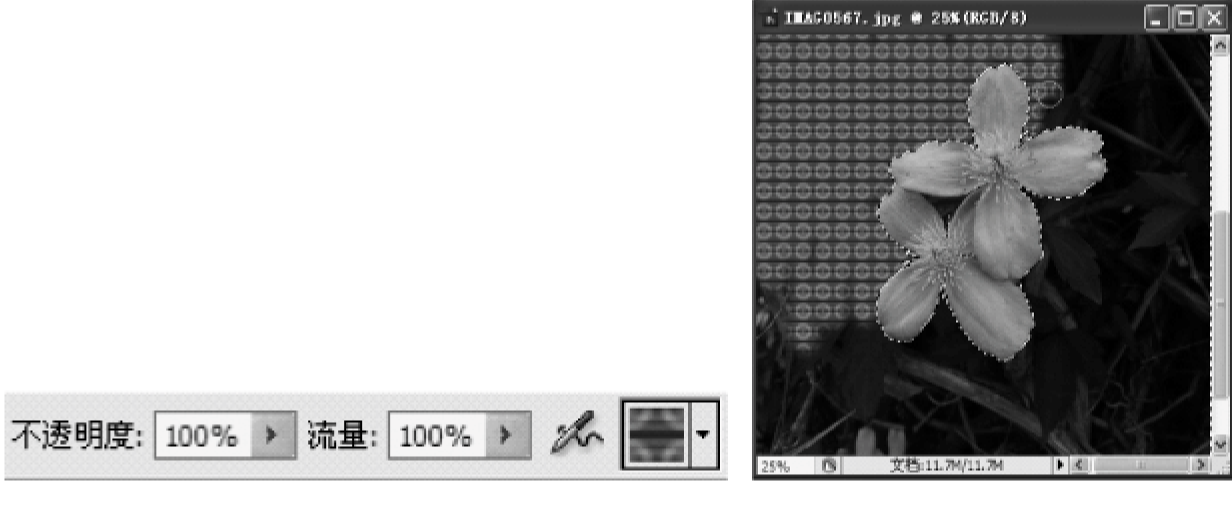


图 5.43　使用图案图章工具复制图像

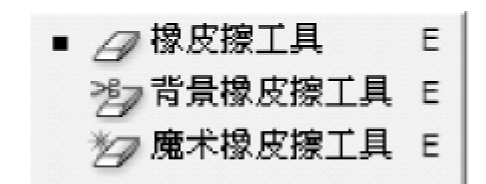
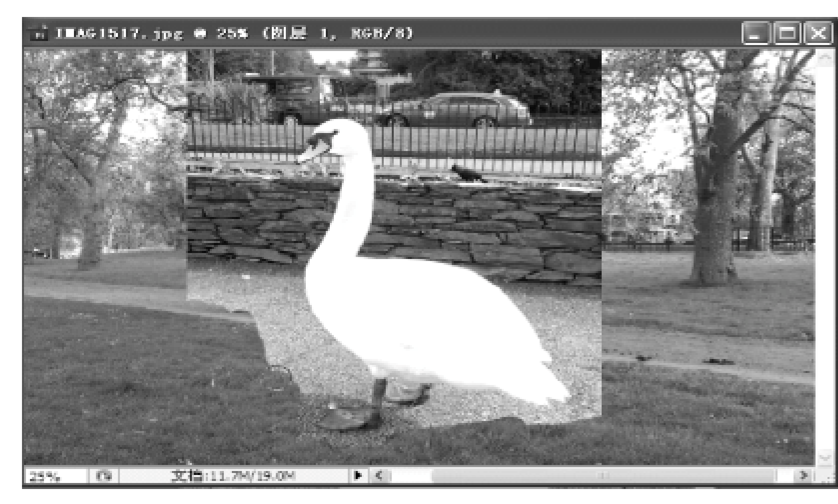


图 5.44　擦除工具

(1) 橡皮擦工具

橡皮擦工具可用来擦除普通图层、背景图层以及选区内的图像。如果擦除的图层是背景图层，会使用背景色颜色替代擦除后的图像区域。如果擦除的不是背景图层，擦除后的图像区域使用透明色替代。使用橡皮擦工具擦除图像如图 5.45 所示。




(a) 擦除普通图层


(b) 擦除背景图层

图 5.45　使用橡皮擦工具擦除图像

在擦除时，只需要单击鼠标左键并移动鼠标即可。在橡皮擦工具的属性栏中可以设置擦除的笔刷大小、不透明度、流量以及抹到历史记录参数。如果选中“抹到历史记录”复选框，需要在历史记录面板中单击鼠标设置还原点，此时，橡皮擦工具就变成图像恢复工具，将图像恢复到设为还原点时操作的效果。橡皮擦工具的属性栏如图 5.46 所示。



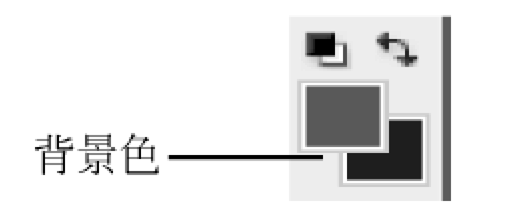
图 5.46　橡皮擦工具的属性栏

（2）背景橡皮擦工具

背景橡皮擦工具将与取样点附近颜色容差相近的区域进行擦除，常用于图像抠取。背景橡皮擦工具会将鼠标单击时的位置设为取样点，鼠标为“+”字光标形状，“+”字光标中心的区域即为擦除区域。取样模式有三种，分别是连续、一次和背景色板。在属性栏中单击对应图标，即可设置取样模式。取样模式设置不同，擦除效果也不同。使用背景橡皮擦工具的 3 种取样模式擦除图像效果如图 5.47 所示。


(a) “连续”取样模式


(b) “一次”取样模式




(c) “背景色板”取样模式

图 5.47　使用背景橡皮擦工具的 3 种取样模式擦除图像

连续：单击鼠标并移动，此时“＋”字光标形状的取样点会随着鼠标的移动而改变，擦除效果比较连续。

一次：单击鼠标并移动，此时会将首次单击鼠标时的位置设为取样点，只要鼠标不松开，取样点就不变化，取样点位置容差相近的颜色就比较容易擦除。即使“＋”字光标中心移到其他位置也不会被擦除。

背景色板：背景橡皮擦工具只对与工具箱中设定的背景色相近的颜色进行擦除。

使用背景橡皮擦工具进行抠图时，在抠取的图像边缘使用“一次”取样模式，设定数值低的容差值，必要时需要在背景橡皮擦工具的属性栏中选中“保护前景色”复选框，只要设置为前景色的颜色都不会被擦除掉(在工具箱中单击吸管工具 ，在需要保护的图像边缘单击鼠标，单击处的颜色自动设为前景色)。只要抠取的图像边缘处理好了，远离边缘的区域可以使用“连续”取样模式或者其他工具进行擦除即可。使用背景橡皮擦工具抠图效果如图 5.48 所示。



图 5.48　使用背景橡皮擦抠图效果

(3) 魔术橡皮擦工具

使用魔术橡皮擦工具进行图像擦除时，首先在属性栏中设置“容差”，然后在擦除的图像区域单击鼠标，与鼠标单击时的取样点颜色相近的图像即被擦除，擦除的区域成“片”状区域。使用魔术橡皮擦工具擦除图像效果如图 5.49 所示。



图 5.49　使用魔术橡皮擦工具擦除图像

7. 图像修复工具

Photoshop 提供的图像修复工具主要用来修复图像，它包括污点修复画笔工具、修复画笔工具、修补工具以及红眼工具。

将鼠标移到工具箱中污点修复画笔工具图标，单击鼠标左键按住 1 秒钟，显示弹出菜单，单击对应图标即可选中该工具，如图 5.50 所示。

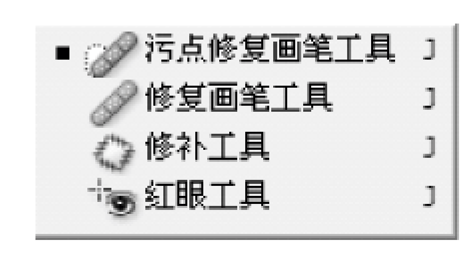


图 5.50　图像修复工具

(1) 污点修复画笔工具和修复画笔工具

污点修复画笔工具和修复画笔工具都可用来修复图像或照片，修复时会将取样点附近的图像复制到有污点的区域，同时取样点处的纹理、光照和阴影也会与当前区域的像素进行匹配，使得修复前后图像的纹理、亮度基本不发生变化。

二者主要区别在于：

使用污点修复画笔工具不需要手动取样，该工具会自动将鼠标单击处的邻近像素作为取样点，操作时只需要将鼠标移到污点区域，单击鼠标左键，移动鼠标即可，操作简单、方便。

使用修复画笔工具需要手动设置取样点（单击鼠标左键的同时按住 Alt 键），然后单击鼠标左键并移动，该工具会将"＋"字光标区域的图像复制到鼠标移动区域，进行图像修复。

使用污点修复画笔工具和修复画笔工具修复图像如图 5.51 所示。




(a) 使用污点修复画笔工具修复图像



(b) 使用修复画笔工具设置取样点



(c) 使用修复画笔工具修复图像

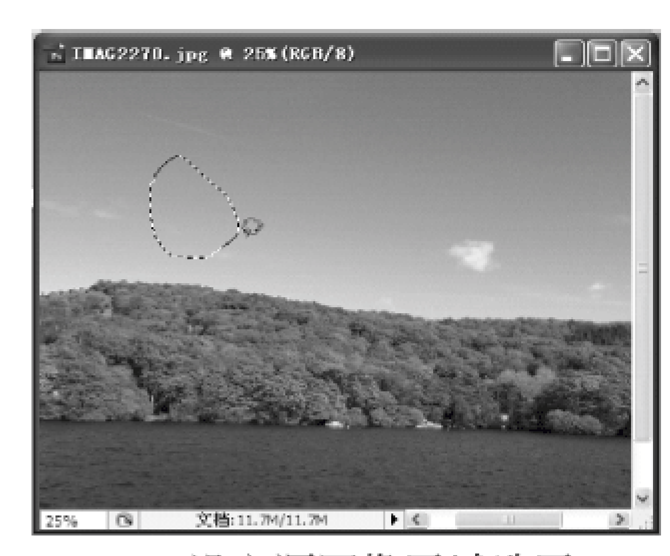
图 5.51　使用污点修复画笔工具和修复画笔工具修复图像

(2) 修补工具和红眼工具

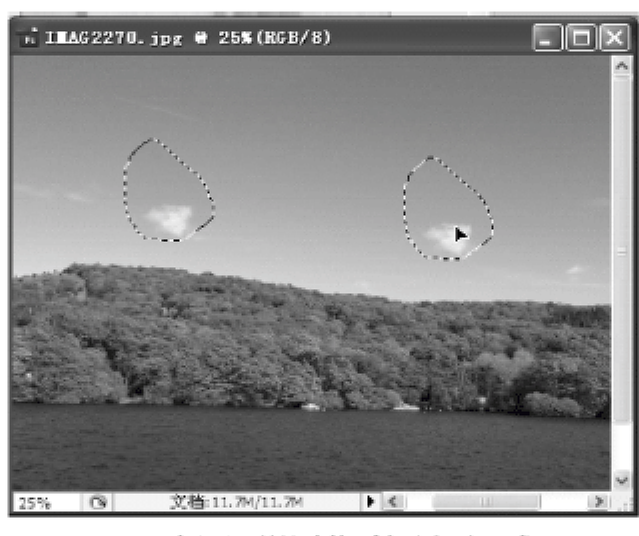
修补工具主要用来修复较大面积的污点。在使用修补工具时，需要单击鼠标左键并拖动，创建源图像区域选区，然后移动鼠标到选区内，单击鼠标左键并移动鼠标到需要修复的

目标区域。

修补工具有两种修补方式：源和目标。需要在修复图像前进行设置，可以通过单击属性栏中“源”和“目标”选项完成，使用修补工具修复图像效果如图 5.52 所示。




(a) 设定源图像区域选区


(b) 采用“源”修补方式

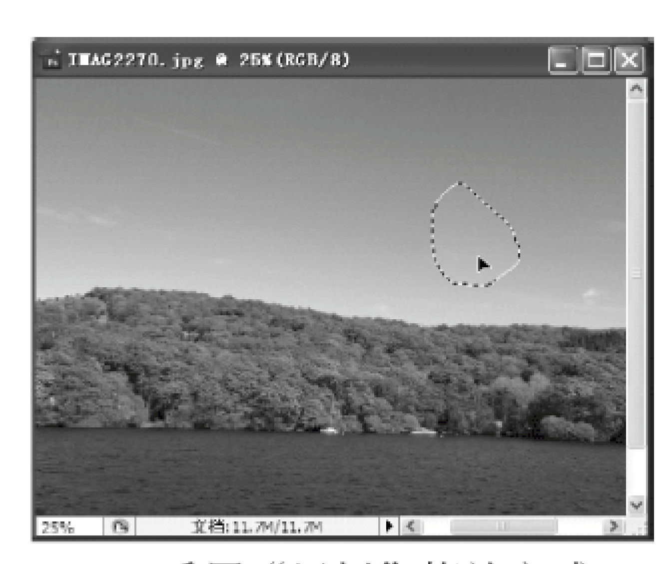

(c) 采用“目标”修补方式

图 5.52　使用修补工具修复图像

源：指的是从目标修补源，是将目标区域的图像替换到源区域。

目标：指的是从源修补目标，是将源区域的图像替换到目标区域。

不管采用何种修补方式，都不是简单的图像替换或复制，而是将替换的图像与当前修复的区域进行纹理、光照和阴影进行匹配，从而使修复后的图像很好地融入图像其余部分。

红眼工具主要用来对照片中的红眼进行修复。为了避免红眼，现在很多数码相机都有红眼去除功能来消除红眼。使用红眼工具需要在属性栏中设置“瞳孔大小”和“变暗量”属性参数。其中，瞳孔大小用来设置红眼工具影响的区域，变暗量则用来校正暗度。

8. 图像修饰工具

使用 Photoshop 提供的图像修饰工具可方便地对图像进行清晰、模糊以及色调、饱和度等细节处理，共包括两组工具：模糊工具、锐化工具、涂抹工具和减淡工具、加深工具、海绵工具。

(1) 模糊工具、锐化工具和涂抹工具

模糊工具、锐化工具和涂抹工具主要用来对图像进行模糊或清晰处理。将鼠标移到工具箱中模糊工具图标，单击鼠标左键按住 1 秒钟，显示弹出菜单，单击对应图标即可选中该工具，如图 5.53 所示。



图 5.53　模糊工具、锐化工具和涂抹工具

模糊工具主要用来将图像变得柔和，起到局部模糊的效果。在属性栏中可以对该工具的“画笔”和“强度”参数进行设置。

画笔：用来设置模糊的大小。

强度：用来设置画笔的力度。数值越大，模糊的效果越强。

锐化工具与模糊工具相反，是将模糊的图像变得相对清晰，无论是模糊还是锐化工具，都会丢失图像信息，如果处理过度，将会导致图像失真。涂抹工具可用来制作手指蘸取湿颜料涂抹的效果。使用模糊工具、锐化工具和涂抹工具处理图像效果如图 5.54 所示。

(2) 减淡工具、加深工具和海绵工具

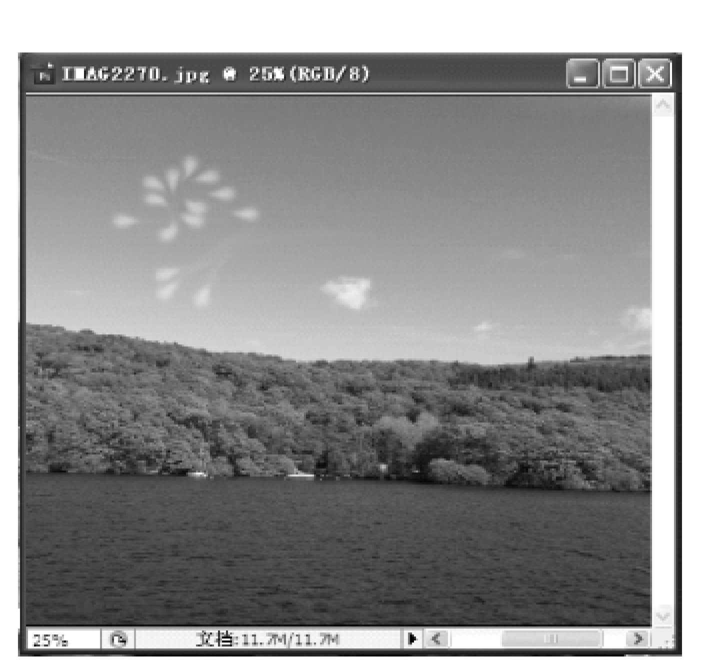
减淡工具、加深工具和海绵工具主要用来对图像的色调和饱和度等细节进行处理。将鼠标移到工具箱中减淡工具图标，单击鼠标左键按住 1 秒钟，显示弹出菜单，单击对应



(a) 模糊工具处理效果



(b) 锐化工具处理效果



(c) 涂抹工具处理效果

图 5.54　使用模糊工具、锐化工具和涂抹工具处理图像

图标即可选中该工具,如图 5.55 所示。

减淡工具可以对图像进行局部减淡,通过提高图像的亮度改善曝光效果,属性栏中“范围”参数用于选择要处理的特殊色调区域,包括“阴影”、“中间调”和“高光”三个选项。



图 5.55　减淡工具、加深工具和海绵工具

阴影:对图像中较暗的区域进行处理。

中间调:对图像整体进行处理。

高光:对图像中高光区域进行处理。

加深工具可以对图像进行局部加深,使用该工具可以加深图像中的阴影或对高光区域进行暗化处理。

海绵工具主要用来改变图像的饱和度,分为加色和去色两种工作模式。加色一般用来增强图像色彩饱和度,使图像颜色更加鲜艳;而去色则是降低图像色彩饱和度,使图像更加阴沉、昏暗。使用减淡工具、加深工具和海绵工具处理图像如图 5.56 所示。



(a) 使用减淡工具增加亮度



(b) 使用加深工具对图像加深



(c) 使用海绵工具“去色”模式



(d) 使用海绵工具“加色”模式

图 5.56　使用减淡工具、加深工具和海绵工具处理图像

5.3　图像基本操作

Photoshop 的图像基本操作主要包括图像打开和保存、图像类型查看、图像移动和复制以及图像变换和图像尺寸调整等，这些操作需要通过菜单实现，下面分别进行介绍。

5.3.1　图像打开和保存

使用 Photoshop 对图像进行处理，首要的就是要打开需要处理的图像文件。选择菜单“文件”→“打开”，在弹出的“打开”对话框中选择要打开的图像文件，单击“打开”按钮，Photoshop 会打开一个新窗口显示该图像文件。

图像在 Photoshop 中处理完成，可以保存成 Photoshop 专用的 Psd 格式，该格式存储了图像在进行处理时用到的图层、蒙版以及色彩等更多细节信息，方便对该图像进行重新编辑修改。

也可以将图像保存成其他图片格式，选择菜单“文件”→“存储为”，输入文件名，在“文件类型”下拉菜单中选择需要保存的图像类型。如果输入的文件名后缀和要保存的图像类型不一致，则会导致该图像无法正确打开。如果选择菜单“文件”→“存储”，则会默认使用当前图像文件类型进行保存。“打开”对话框和“存储为”对话框如图 5.57 所示。





图 5.57　“打开”对话框和“存储为”对话框

5.3.2　图像类型查看

选择菜单“图像”→“模式”，在弹出的菜单中显示✔图标的选项即是当前图像的类型信息，如图 5.58 所示。如果要更改当前图像类型，只需要在图像类型处单击鼠标左键。

Photoshop 支持的颜色模式包括灰度、索引颜色、RGB 颜色、CMYK 颜色、Lab 颜色以及多通道模式。

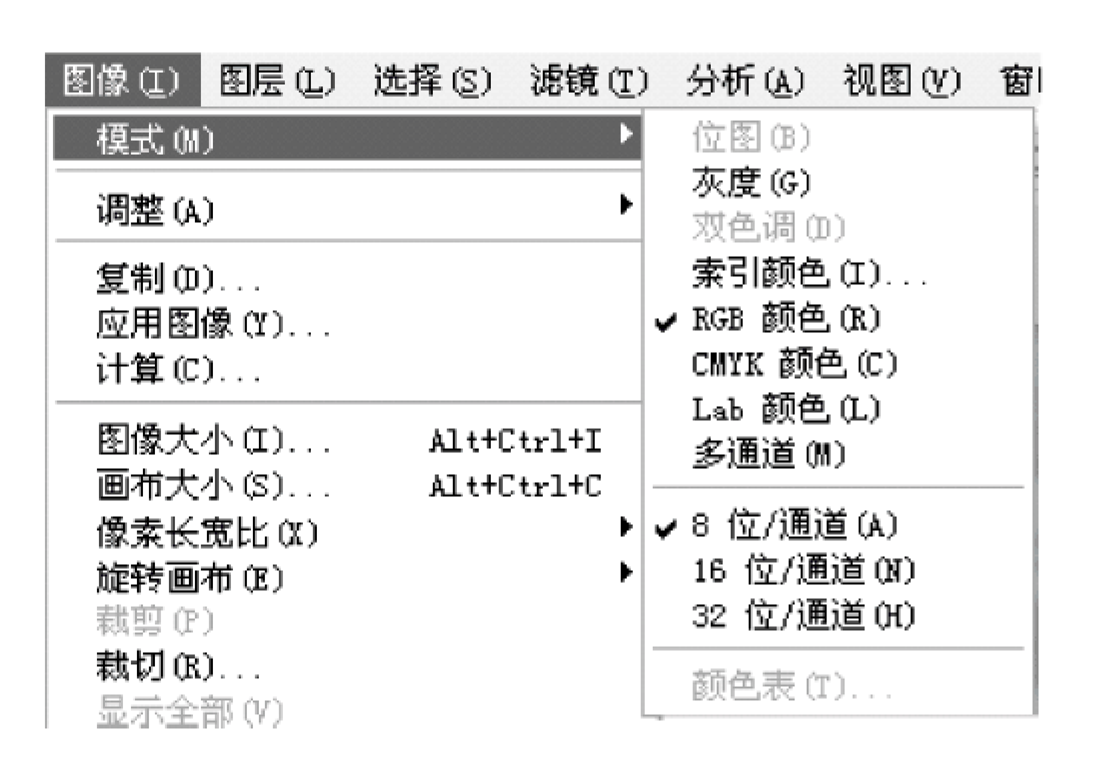


图 5.58 查看图像类型

• 灰度模式

该模式可表示纯白、纯黑以及白到黑之间的过渡色，可使用多达 256 级灰度来表现图像。灰度图像的每个像素有一个 0(黑色)到 255(白色)之间的亮度值。灰度值也可以用黑色油墨覆盖的百分比来度量(0 等于白色，100%等于黑色)。

• 索引颜色

也称为映射颜色，该模式采用一个颜色表存放并索引图像中的颜色，可最多生成 256 种颜色的 8 位图像文件，不适合进行复杂的图像编辑和操作。在 Photoshop 中如果当前图像类型是索引颜色，则不允许进行滤镜等图像处理操作。

• RGB 颜色

该模式是一种加色的彩色模型，是计算机显示彩色图像的最佳模式。在 RGB 彩色模型中，任何一种 RGB 颜色都是由红(R)、绿(G)、蓝(B)这三种颜色分量按照不同比例混合而成。每种颜色分量由 8 位二进制表示，RGB 彩色模型共可以表示出 $2^8\times2^8\times2^8=2^{24}=$ 16 777 216 种颜色，简称为 1600 万色或 24 位色。RGB 色彩模式的值是 0～255 之间的整数值。例如当红、绿、蓝都是 255 时，构成白色，当红、绿、蓝都是 0 时，构成黑色。

• CMYK 颜色

该模式也称为印刷色彩模式，是一种复杂的彩色模型，主要应用在印刷行业。与 RGB 彩色模型类似，CMYK 颜色模型是由四种油墨色构成，分别是青(C)、品红(M)、黄(Y)以及黑色(K)。每种油墨色由 8 位二进制表示，CMYK 彩色模型共可以表示出 $2^8\times2^8\times2^8\times2^8=2^{32}=$4 294 967 296 种颜色。CMYK 色彩模式的值是由添加油墨浓度的百分比来表示，例如暗红色是由 20%青、100%品红、80%黄和 5%黑构成。当四种基色值均为 0 时，就会产生纯白色。

• Lab 颜色

该模式是一种不依赖光线及颜料的颜色模式，包括亮度 L(0～100)、绿色到紫红色的色彩 a(+127～-128)、蓝色到黄色的色彩 b(+127～-128)三种通道。Lab 描述的是颜色的显示方式，而不是设备(如显示器)生成颜色所需的特定色料的数量，所以 Lab 被视为与设备无关的颜色模型。有时候 Lab 颜色模式被用作 CMYK 模式和 RGB 模式转换的桥梁。

• 多通道

该模式图像在每个通道中包含 256 个灰阶，对于特殊打印很有用。例如，如果图像中只使用一两种或两三种颜色，使用多通道模式可以减少印刷成本并保证图像颜色正确输出。

5.3.3 图像剪切和复制

要想对图像进行剪切操作，首先使用选区工具将需要剪切的图像区域设定为选区，接下来在工具箱中单击 移动工具图标，将鼠标移到选区内，此时鼠标变成 形状，然后单击鼠标左键移动鼠标，即可将选区内的图像剪切。如果在单击鼠标移动鼠标的同时按住 Alt 键，此时鼠标变成 形状，即可将选区内的图像进行复制，如图 5.59 所示。

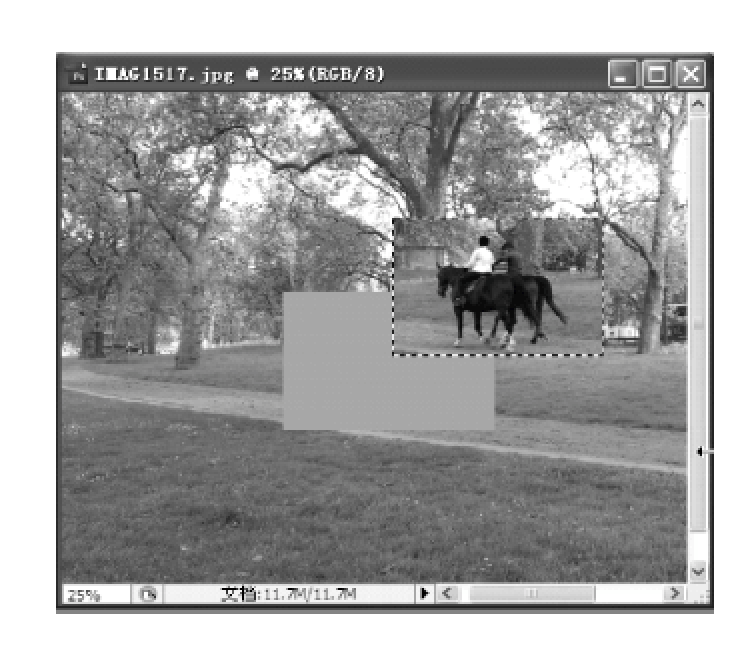




图 5.59 图像剪切和复制

也可以使用 Ctrl+C 键将选区内的图像复制到剪贴板上，使用 Ctrl+V 键将剪贴板上的图像复制到指定图像文件中，Photoshop 会自动创建新图层存储复制的图像。

5.3.4 图像形状变换

图像形状变换包括缩放、旋转、斜切、扭曲、透视和变形。进行变换前，首先使用选区工具将需要进行形状变换的图像区域创建为选区，然后选择菜单“编辑”→“变换”，在弹出的菜单中选择要应用的变换选项，此时选区四周出现带有锚点的矩形控制框，如图 5.60 所示。



图 5.60 带有锚点的矩形控制框

1. 缩放

将鼠标移到矩形控制框上的锚点，鼠标变成 ↔ 形状，单击鼠标左键向里或向外任意拖

曳,对选区内的图像放大和缩小。

2. 旋转

将鼠标移到矩形控制框任意角上,鼠标变成↰形状,单击鼠标左键按顺时针或逆时针方向移动鼠标,对选区内的图像进行旋转。

3. 斜切

对矩形区域进行斜切变换就是对其进行平行四边形变换。将鼠标移到矩形控制框的水平或垂直边框,鼠标变成▶↔或▶↕形状,单击鼠标左键左右或上下移动鼠标,对选区内的图像斜切变换。

4. 扭曲

将鼠标移到矩形控制框任意角上,鼠标变成▶形状,单击鼠标左键任意拖动角上的锚点,对选区内的图像进行扭曲变换。

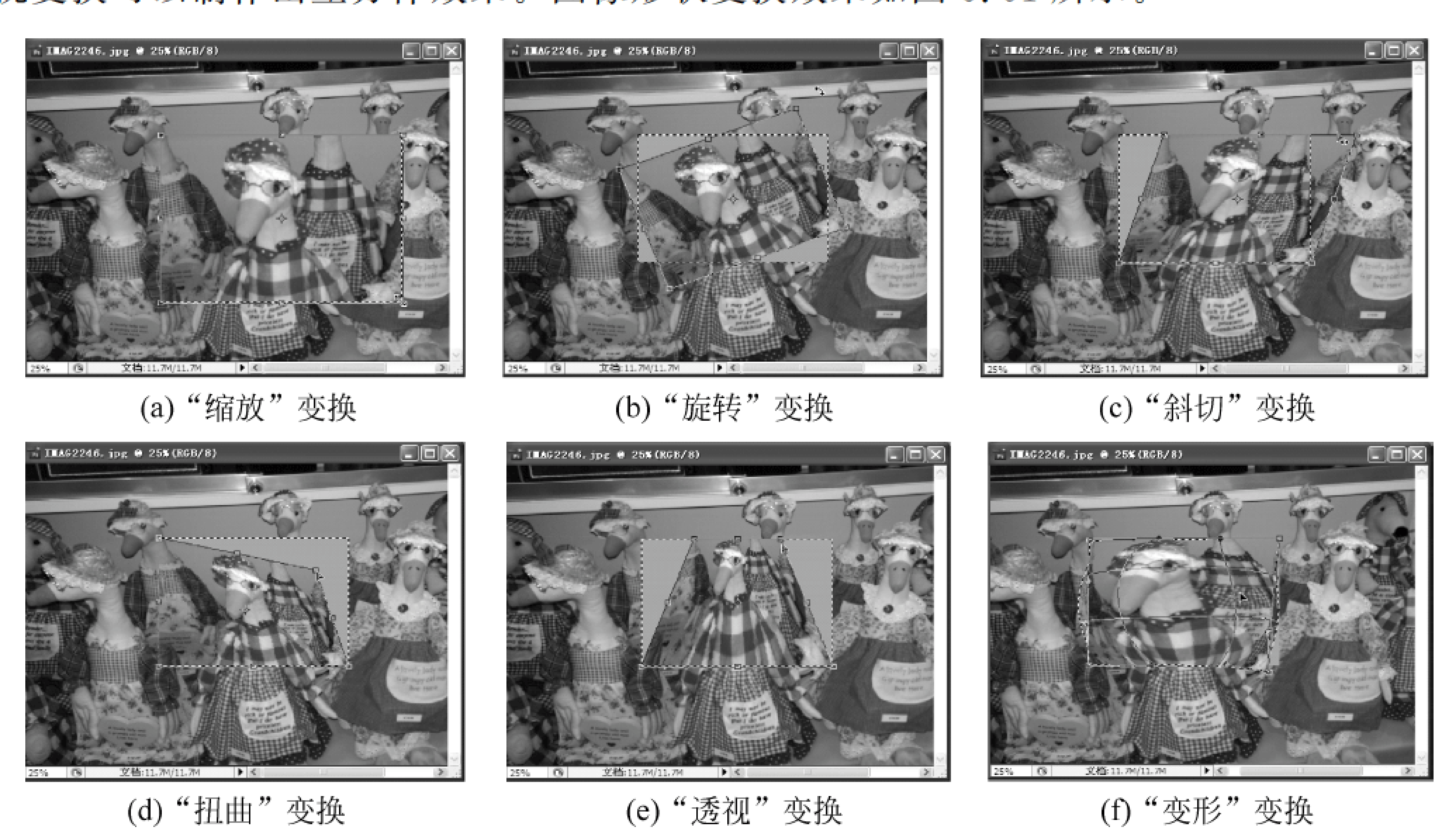
5. 透视

对矩形区域进行透视变换就是对其进行梯形变换。将鼠标移到矩形控制框任意角上,鼠标变成▶形状,单击鼠标左键向里或向外拖拽鼠标,对选区内的图像透视变换。

6. 变形

将鼠标移到矩形控制框以内任意位置,鼠标变成▶形状,单击鼠标左键并拖动,对选区内图像进行任意变形。

如果想退出形状变换的编辑状态,可以单击工具箱中任意图标。通常使用斜切、扭曲和透视变换可以制作出立方体效果。图像形状变换效果如图 5.61 所示。



(a)"缩放"变换 (b)"旋转"变换 (c)"斜切"变换

(d)"扭曲"变换 (e)"透视"变换 (f)"变形"变换

图 5.61 图像形状变换

5.3.5　图像翻转变换

图像翻转变换包括水平翻转和垂直翻转。水平翻转可制作出图像左右对称效果，垂直翻转可制作出水面镜像效果。在进行翻转变换前，首先使用选区工具将需要变换的图像创建为选区，然后选择菜单“编辑”→“变换”→“水平翻转”或“编辑”→“变换”→“垂直翻转”，选区内的图像自动翻转。

5.3.6　图像色相、饱和度和明度调整

色相、饱和度和明度构成了色彩的三要素。色相是指色彩的相貌，如红、黄、绿、蓝等各有自己的色彩相貌，是区别各种不同色彩的主要依据。饱和度是指色彩的纯度，其值越高，表现越鲜明，其值越低，表现越暗淡。明度是指色彩的明暗程度，白色明度最高，黑色明度最低。

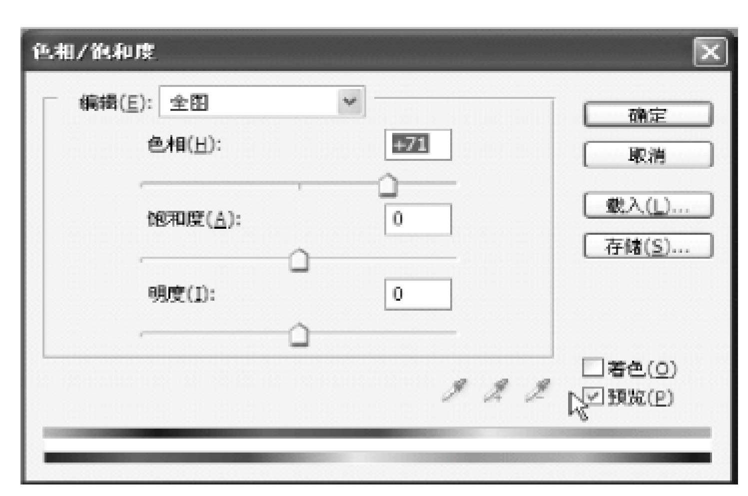
在 Photoshop 中，对图像的色彩调整实际上就是对色彩三要素进行调整的。首先选择菜单“图像”→“调整”→“色相/饱和度”，在弹出的“色相/饱和度”对话框中移动色相、饱和度和明度的滑块来对其进行更改。在“色相/饱和度”对话框下方，靠近上面的颜色条是未更改数值的颜色分布，靠近下面的颜色条则是更改完数值之后的颜色分布，通过对比两个颜色条的变化，可以很容易看出更改前后颜色的变化。图像色相调整效果如图 5.62 所示。



(a) 原图



(b) 调整之后



(c) “色相/饱和度”对话框

图 5.62　调整图像色相

单击“色相/饱和度”对话框中的“编辑”参数下拉菜单选项，可以用来设置对色相、饱和度和明度起作用的色彩范围，这只是大概的色彩范围。如果只对图像中指定的颜色进行色相、饱和度和明度的更改，可以设置对话框下方的吸管工具 、添加到取样工具 和从取样中减去工具 。吸管工具 可以用来将图像中指定的颜色设置为编辑的色彩范围，只对该范围的颜色进行更改，图像中的其他颜色不受影响；添加到取样工具 可用来扩展当前编辑的色彩范围；从取样中减去工具 可用来减少当前编辑的色彩范围。

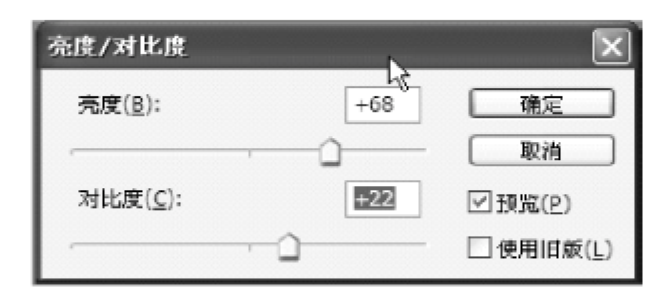
5.3.7　图像亮度和对比度调整

亮度是指光线的明亮程度，可通过调整亮度改善曝光效果。对比度是指颜色的对比程度，对比度值增加可在一定程度上增加图像清晰程度，过度则会产生失真。在 Photoshop 中，如果想对图像的亮度和对比度调整，可以选择菜单“图像”→“调整”→“亮度/对比度”，如

图 5.63 所示。


(a) 原图


(b) 调整之后



(c) “亮度/对比度”对话框

图 5.63 图像“亮度/对比度”调整

5.3.8 图像尺寸调整

在 Photoshop 中,可以通过更改画布大小和图像大小改变图像尺寸。

1. 更改画布大小

在 Photoshop 中,图像都是在画布上进行显示和处理的。画布大小其实就是图像文档的尺寸大小,直接影响图像文件的大小。当画布不足以显示更多的图像内容时,可以更改画布大小。

选择菜单“图像”→“画布大小”,在弹出的“画布大小”对话框中设置新画布的“宽度”和“高度”尺寸。“定位”参数则用来定义新画布的扩展方向,“白色”方形按钮代表当前画布位置,四周的箭头表示画布扩展的方向。在空格中单击鼠标左键,就可以设置当前画布在新画布中的位置。

当新画布尺寸小于当前画布时,图像上部分显示内容会被裁掉,图像信息丢失。当新画布尺寸大于当前画布时,新增加的画布可以显示更多的图像,原有图像不受影响。更改图像的画布大小如图 5.64 所示。

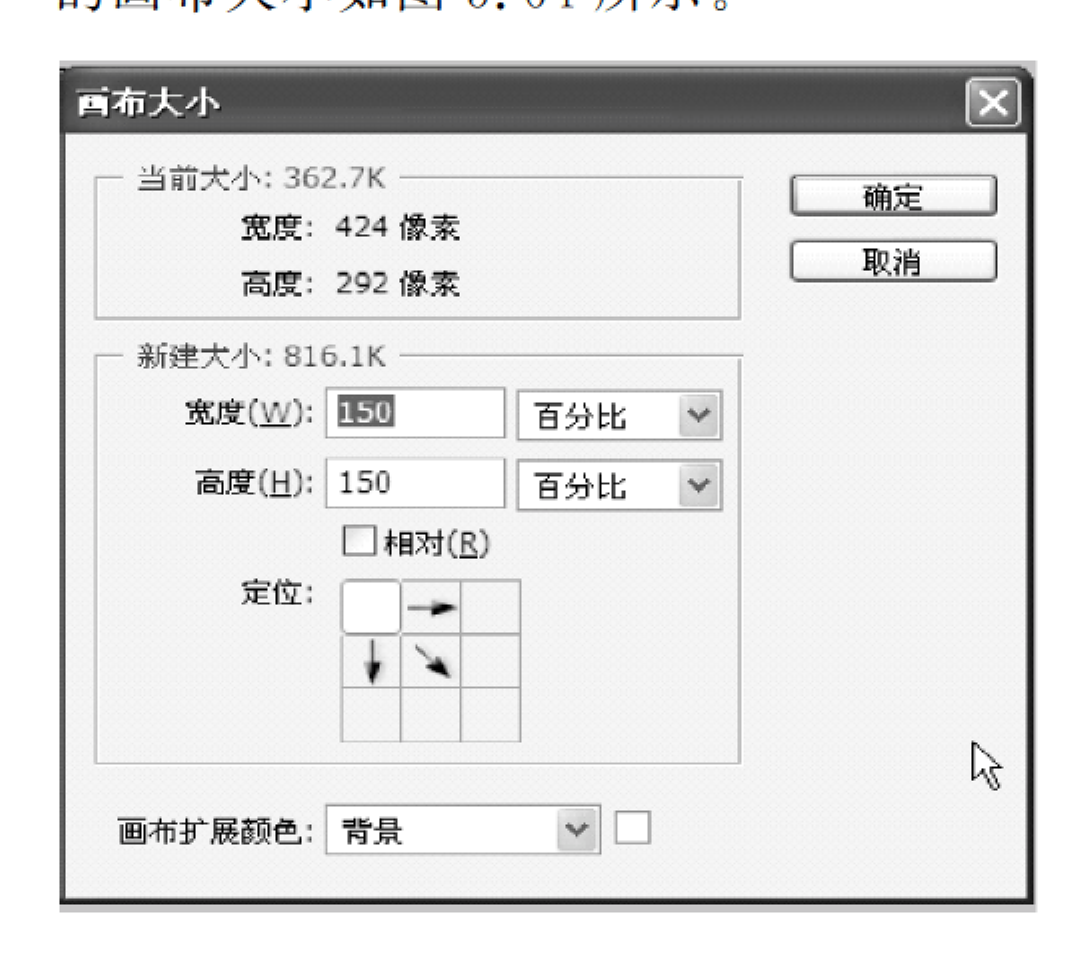




图 5.64 更改图像的画布大小

2. 更改图像大小

选择菜单“图像”→“图像大小”，弹出“图像大小”对话框，可以更改像素大小，也可以更改文档大小。像素大小是指图像文件显示的像素多少，变大变小会直接影响图像的显示效果。文档大小是指图像文档尺寸大小。如果取消“图像大小”对话框下方的“重定图像像素”复选框，则图像像素大小不变，只能改变文档大小，值越大分辨率越低，图像越模糊；值越小分辨率越高，图像越清晰。“图像大小”对话框如图 5.65 所示。

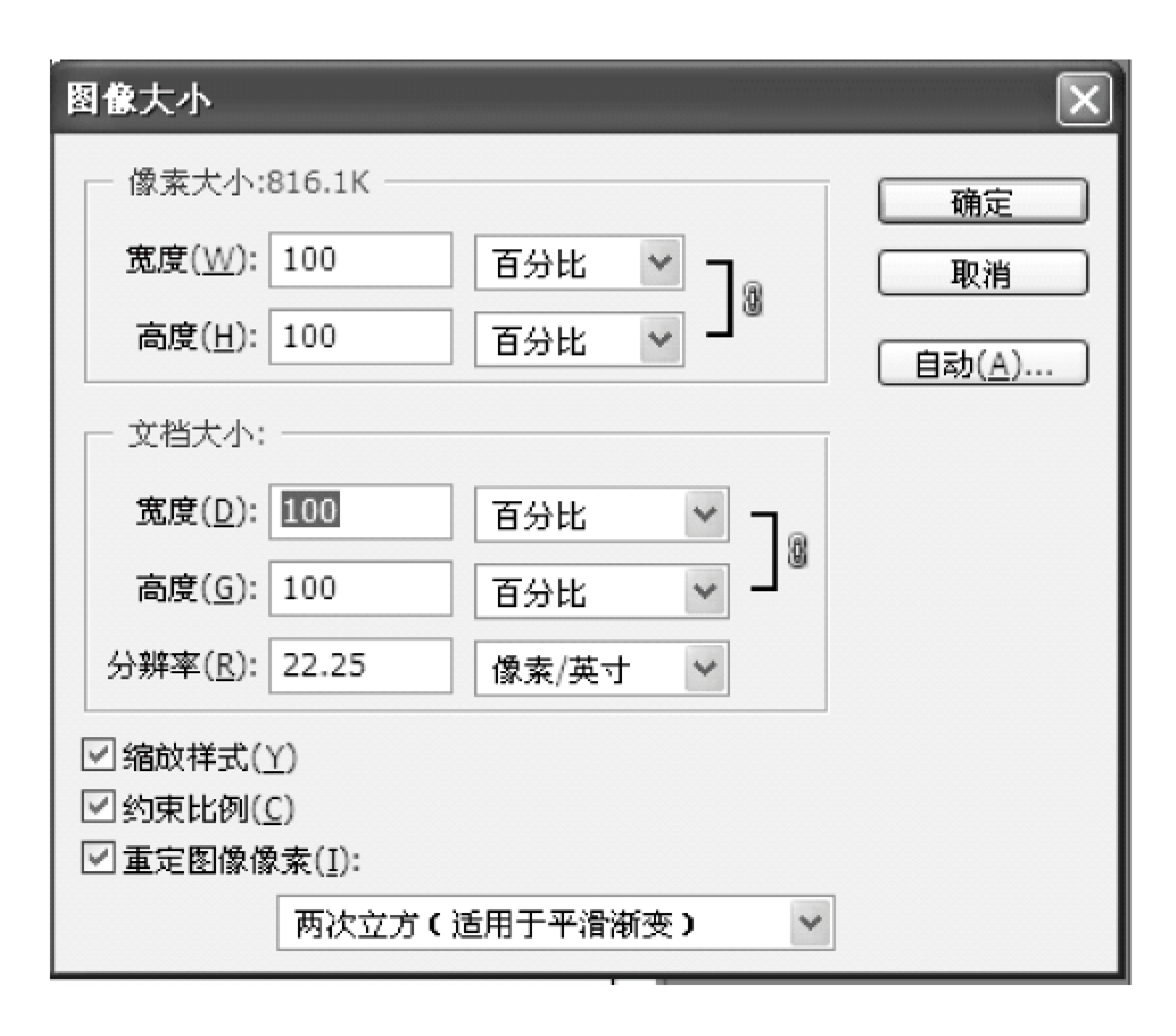


图 5.65 “图像大小”对话框

提示：像素是构成位图的基本单位，也称为像点，图像中细节的数量取决于其像素多少。而影响图像清晰程度的是图像分辨率，即每英寸显示的像素个数，使用水平像素×垂直像素表示。

5.4 图层和滤镜

使用 Photoshop 处理图像的最大优势是图层，使用图层可以很方便地修改图像，简化图像编辑操作，还可以创建各种图层特效，从而制作出各种特殊效果。

滤镜是一组完成特定效果的程序，一个滤镜对应一种特定的效果，可用来实现图像的各种特殊效果。下面分别进行介绍。

5.4.1 图层

图层可以被看做一张背景透明的画纸，将图像的不同部分通过图层进行存放，多个图层上下叠加在一起从而组合成复杂图像。使用图层的优点是：单独对某个图层进行操作不会影响到图像中的其他图层，

图层的操作可以通过图层面板和图层菜单来完成的。图层面板显示了当前图像的所有图层，如图 5.66 所示。

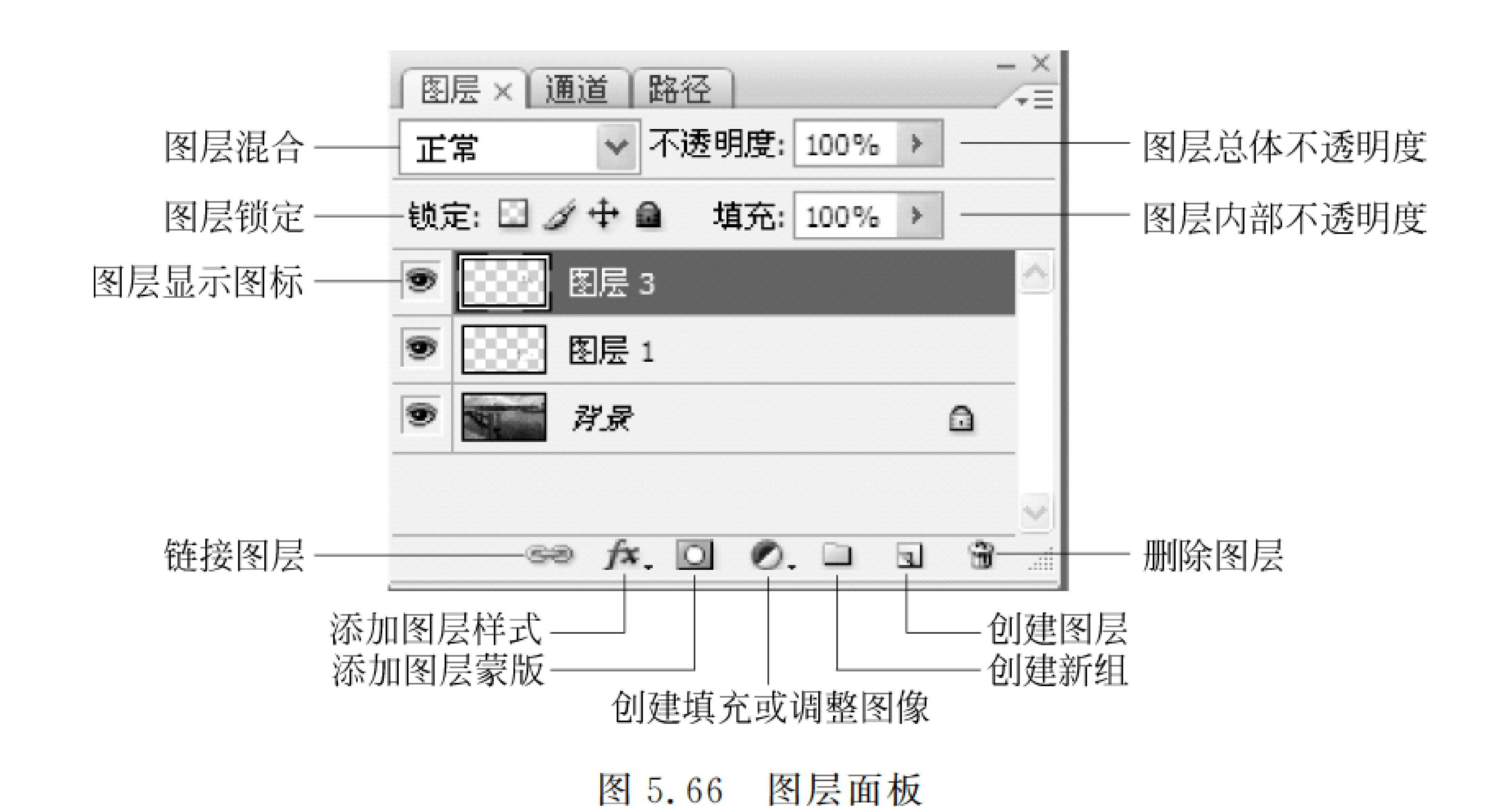


图 5.66　图层面板

1. 图层类型

Photoshop 中定义了多种类型的图层，例如背景图层、文本图层、调节图层和形状图层等。不同类型的图层，特点和功能不相同，而且操作和使用方法也不尽相同。下面就来介绍各种图层类型。

(1) 普通图层和背景图层

普通图层是 Photoshop 最常用的图层，Photoshop 的大部分图像处理功能都可以对该类型的图层实现。

单击图层面板的 按钮或选择菜单“图层”→“新建”→“图层”，弹出“新建图层”对话框，设置图层“名称”、“不透明度”以及“模式”参数后，单击“确定”按钮，就可以创建普通图层。

背景图层一般位于图层面板的最底部，以斜体字“背景”标识，所有新打开的图像都是作为背景图层显示在图像窗口中的。背景图层是被锁定的，图层面板中的图层混合模式、图层不透明度、填充不透明度以及锁定属性不能用来对背景图层进行设置或更改。

如果要更改背景图层的不透明度和图层混合模式等，需要将其转换为普通图层。选择菜单“图层”→“新建”→“背景图层”，在弹出“新建图层”对话框中设置图层“名称”、“不透明度”以及“模式”参数后，单击“确定”按钮，即可将其转换为普通图层，如图 5.67 所示。





图 5.67　将“背景图层”转换为“普通图层”

（2）文本图层

当使用 T 横排文字工具和 IT 直排文字工具在图层上添加文字时，Photoshop 会自动创建文本图层。文本图层可以执行“缩放”、“旋转”、“斜切”、“变形”等操作，如果要想执行扭曲、透视或滤镜操作，需要对其执行“栅格化”操作。

在图层面板中单击鼠标选中要处理的文字图层，然后选择菜单“图层”→“栅格化”→“文字”，即可将文字图层转换成普通图层。如果直接对文字执行滤镜操作，Photoshop 也会弹出提示对话框，单击“确定”按钮，也可以将其转变为普通图层。文字图层如图 5.68 所示。



图 5.68　文字图层

（3）调整图层和填充图层

调整图层和填充图层是比较特殊的图层，分别存储了色彩、色调的调整和填充信息，主要用来调整图像的色彩和填充效果。调整图层和填充图层只对位于它下方的图层起作用，通过图层叠加显示图像最终修改效果。由于没有改变原图像信息，可以随时恢复图像。

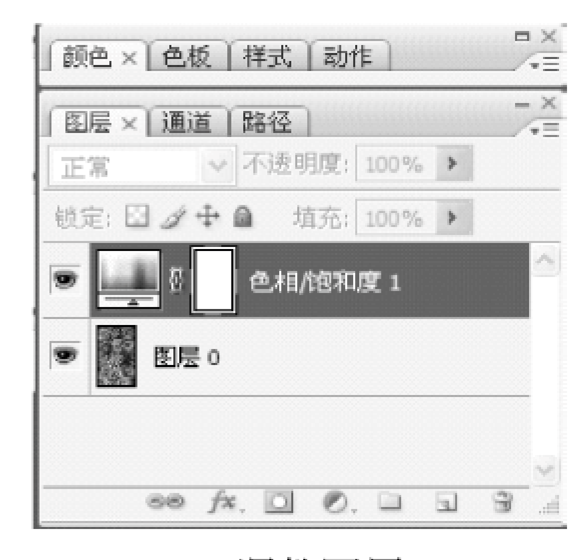
使用调整图层修改图像饱和度的方法是：在图层面板中选择要处理的图层，然后选择菜单“图层”→“新建调整图层”→“色相/饱和度”，在弹出的“新建图层”对话框设置新建调整图层的“名称”、“不透明度”等参数，然后单击“确定”按钮，在弹出“色相/饱和度”对话框，设置“饱和度”值为＋67，单击“确定”按钮。创建“色相/饱和度”调整图层修改图像效果如图 5.69 所示。



(a) 调整前



(b) 调整后



(c) 调整图层

图 5.69　创建“色相/饱和度”调整图层修改图像

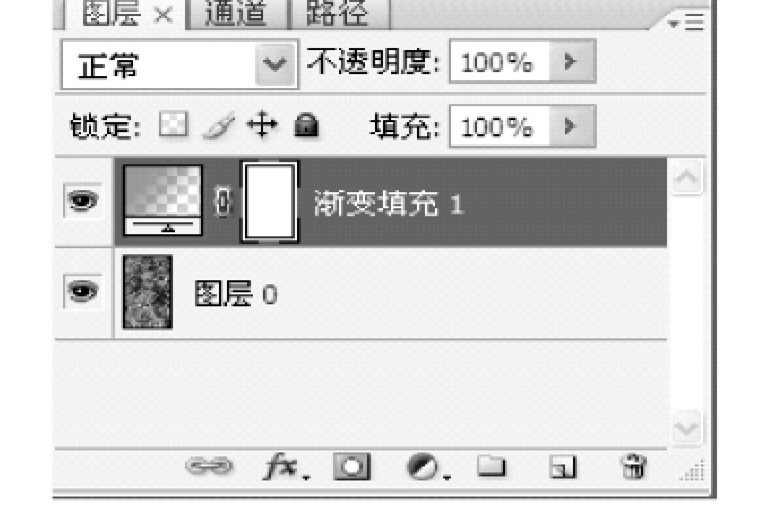
使用填充图层进行渐变填充的方法是：在图层面板中单击要处理的图层，然后选择菜单“图层”→“新建填充图层”→“渐变”，在弹出的“新建图层”对话框设置新建调整图层的“名称”、“不透明度”等参数，然后单击“确定”按钮，在弹出“渐变填充”对话框中设置相应的参数值，单击“确定”按钮。创建“渐变”填充图层修改图像效果如图 5.70 所示。

创建其他调整和填充图层方法类似，这里不再一一描述。

2. 图层操作

图层操作包括图层的移动/复制/删除、图层锁定、图层叠放次序调整、图层链接、图层合


(a) 使用渐变色填充后



(b) 填充图层

图 5.70 创建“渐变”填充图层修改图像

并以及图层混合等操作，下面分别介绍。

（1）图层移动、复制和删除

图层移动、复制和删除是图层最基本的操作，在使用 Photoshop 处理图像时会经常用到。在进行图层移动、复制和删除前，要首先在图层面板中单击鼠标左键选择要操作的图层。

如果想要移动图层，单击工具箱中的移动图标，然后将鼠标移到目标图层所在区域，单击鼠标左键拖动鼠标，将图层移到适合的位置。

如果想要复制图层，选择菜单“图层”|“复制图层…”，弹出“复制图层”对话框，“为”参数用于设置复制后的图层名称。“文档”参数用于设置复制后的图层存储的文件，下拉列表框自动列出当前在 Photoshop 中已打开的所有图像文件，如果在下拉列表框中选择“新建”，并在“名称”文本框中输入新文件名，Photoshop 会新建图像文件以存储复制后的图层。“复制图层”对话框如图 5.71 所示。

如果想要删除图层，只需要将选择的图层拖动到图层面板下方的垃圾箱图标位置即可。



图 5.71 “复制图层”对话框

（2）图层锁定

为了防止对其他图层处理时对已完成图层进行误操作，可以在图层面板中将已完成图层设置为锁定。Photoshop 提供了四种类型的锁定，包括锁定透明像素、锁定图像像素、锁定位置和全部锁定。

锁定透明像素：对图层中的透明部分进行锁定，此时只能对图像中有像素的部分进行编辑。

锁定图像像素：对图层所有部分都不允许进行编辑，包括透明部分和有图像部分。

锁定位置：对图层位置进行锁定，不允许移动。

全部锁定：完全锁定图层，此时图层面板上的图层混合模式、不透明度、删除图层等功能均不能使用。

锁定图层方法是：首先在图层面板中选择要锁定的图层单击鼠标左键，然后在图层面板中单击对应锁定图标，即可设置锁定类型，同时会在该图层的右方出现 图标。如果要解除锁定，只需再次单击相应的锁定图标即可，如图 5.72 所示。

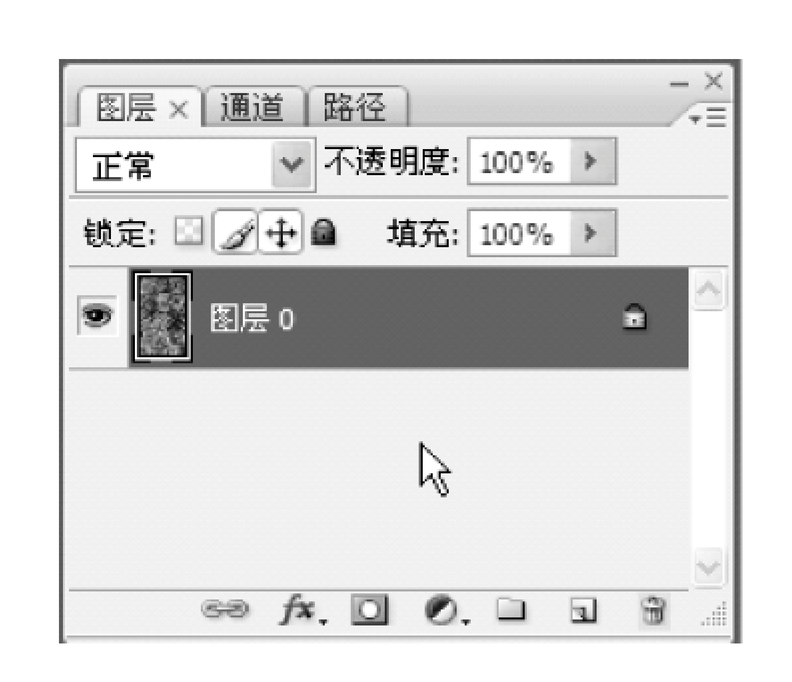


图 5.72　图层锁定

（3）图层叠放次序调整

通常情况下，最先建的图层位于图层面板的底部，最后建的图层位于最上层，上面的图层总会遮住下面的图层，因此，有时候需要通过调整图层的叠放次序，调整图层的显示效果。

调整图层叠放次序的方法是：在图层面板中单击鼠标左键，选中要调整的图层，然后单击鼠标左键将图层拖动到合适位置，如图 5.73 所示。



(a) 原图



(b) 调整后



(c) 在“图层”面板调整图层叠放次序

图 5.73　调整图层叠放次序

（4）图层链接和合并

如果想同时对多个图层进行移动、翻转、自由变换和旋转操作，可以设置多个图层链接，链接后的图层可以执行相同操作。

在图层面板中，单击鼠标左键并按下 Shift 键以选中要链接的图层，然后单击图层面板下方的链接图标 ，此时链接图层右方显示 图标。如果想解除链接，选中链接的图层，然后再次单击图层面板下方的链接图标 即可，如图 5.74 所示。





图 5.74　链接图层

在处理图像时，如果不再需要对图层进行修改，可以将图层合并，节省缓存盘上的空间，提高 Photoshop 程序的运行速度。

Photoshop 支持三种合并方式：向下合并、合并可见图层和拼合图层。

向下合并：将当前图层和其下一层图层进行合并。

合并可见图层：将所有标识为 图标的可见图层进行合并，隐藏图层不合并。



图 5.75　合并图层菜单

拼合图层：合并所有图层，如果有隐藏图层，弹出对话框提示。

在图层面板中选择想要合并的图层，然后选择菜单“图层”→“向下合并”或“图层”→“合并可见图层”或“图层”→“拼合图层”，执行图像合并操作，如图 5.75 所示。

（5）图层混合

在混合图层时，图层面板中的不透明度、填充和混合模式属性可用来制作与众不同的图像合成效果。

不透明度：设置图层总体不透明度。

填充：设置图层内部不透明度。

混合模式：指当前图层与其他图层叠加时，不同图层中的像素如何进行混合。Photoshop 存储了 24 种混合模式，可以通过混合模式的下拉列表框进行选择。

3. 图层蒙版

图层蒙版是进行图像合成时常用的技术，使用该技术在不更改原图的基础上控制图层的部分显示或者隐藏。在蒙版中，黑色部分表示隐藏图层中的图像，白色部分表示显示图层中的图像，渐变色部分则表示逐渐显示图层中的图像。

为图层添加蒙版方法是：在图层面板中，首先选择要应用蒙版的图层，然后单击图层面板下方的 图标，选中的图层右边会显示一个矩形框，该矩形框表示添加的蒙版。接下来在图层面板中选择左边的图层图标，当矩形图标四周出现边框，表示被选中。使用选区工具在图像窗口将需要隐藏的区域创建为选区，然后在图层面板中选择当前图层右边的蒙版图标，并在工具箱中选择画笔工具，设置前景色为黑色，在选区内绘图，选区内的图像即会消失。如果设置画笔颜色为白色进行绘图，则会将之前消失的图像恢复出来。使用图层蒙版隐藏图像如图 5.76 所示。





图 5.76　使用图层蒙版隐藏图像

如果要删除添加的蒙版，只需要在图层面板中选中要删除的蒙版图标，然后单击图层下方的垃圾桶图标，弹出删除蒙版提示窗口，单击“删除”按钮，即可删除蒙版而不会影响原来图层。删除蒙版提示窗口如图5.77所示。

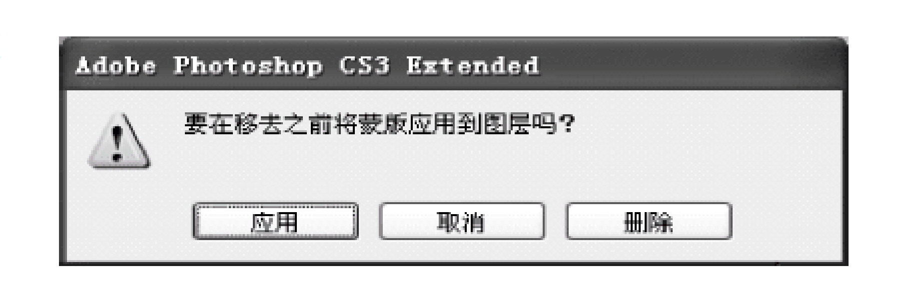


图5.77　删除蒙版提示窗口

4. 图层样式

为图层添加图层样式，可以创建出特殊的修饰效果。Photoshop提供了“阴影”、“内发光”、“外发光”、“斜面和浮雕”等样式，介绍如下。

(1)“投影”和“内阴影”样式

可以增加图层的层次感，通常应用到文字、按钮以及边框的制作中。投影是添加图层边缘以外的阴影效果，内阴影则是添加图层边缘以内的阴影效果。

(2)“内发光”和“外发光”样式

可以对图层边缘添加发光效果。内发光是对图层边缘以内添加发光效果。在内发光的属性中增加了“源”，用来设置中心发光或边缘发光。外发光是对图层边缘以外添加发光效果。

(3)“斜面和浮雕”样式

可以为图层的边缘添加高光和暗调，从而在图层的边缘产生立体的斜面效果或浮雕效果。

(4)“光泽”样式

可以为图层的上方添加波浪形或者绸缎效果。

(5)“颜色叠加”、“渐变叠加”和“图案叠加”样式

可以为图层添加颜色、渐变色以及图案的叠加效果。

(6)“描边”样式

使用纯色、渐变色或图案为图层边缘描绘轮廓。

为图层添加图层样式的方法是：首先在图层面板中选择图层，然后单击图层面板下方的 fx. 图标，弹出图层样式菜单，单击鼠标左键选择要应用的样式，然后在弹出的“图层样式”对话框中设置相应属性参数即可。此时，图层面板中该图层下面会显示出添加的样式名称，如图5.78所示。

如果要修改图层样式，需要在图层面板中双击图层样式名称，在弹出的“图层样式”对话框中更改设置。如果要删除图层样式，同样在图层面板中单击图层样式，然后单击图层面板下方的垃圾桶图标 即可。

在图5.78中，“图层样式”对话框中的“投影”样式参数介绍如下：

混合模式：阴影和图层混合显示的效果，在其右侧有一颜色框，单击设置投影颜色。

不透明度：阴影的不透明度，值越大阴影颜色越深。

角度：光线照明角度，阴影方向会随光照角度的变化而发生变化。

使用全局光：为同一图像的所有图层样式设置相同的光线照明角度。

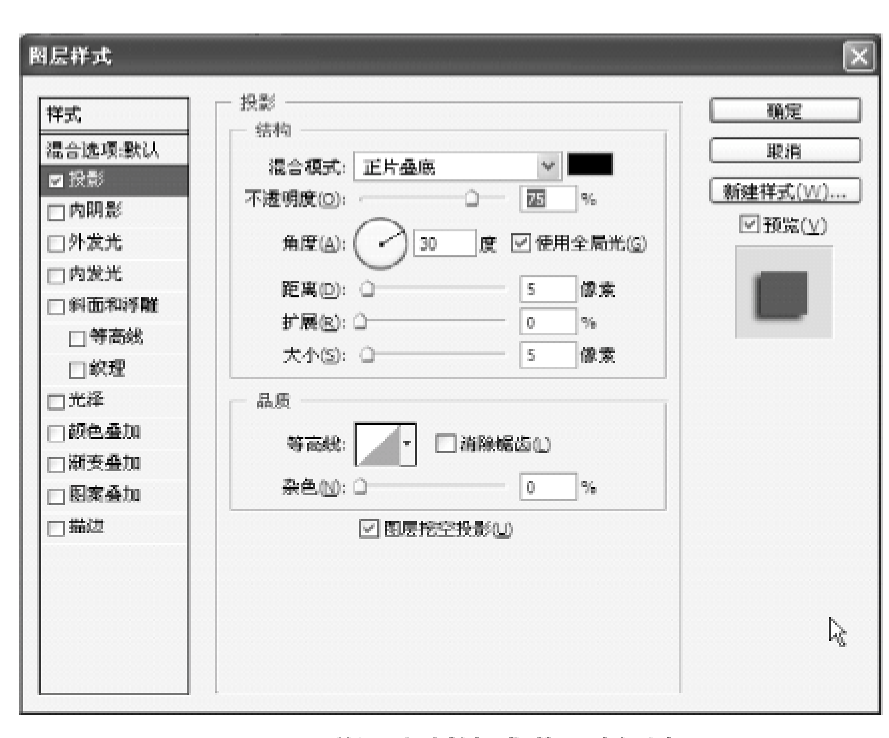
距离：阴影与图层的距离。

扩展：光线的强度，值越大，投影效果越强烈。





(a) 图层样式菜单



(b) “图层样式”对话框



(c) 图层面板中图层样式

图 5.78　图层样式设置

大小：阴影柔化程度，值越大，柔化程度越大。

等高线：用来对阴影部分做进一步设置。单击弹出“等高线”编辑器，等高线的高处对应阴影上的暗圆环，低处对应阴影上的亮圆环。

杂色：对阴影部分添加随机的透明点。

图 5.79 给出了对文字图层添加的“投影”、“浮雕”和“渐变叠加”样式。



(a) “投影”样式效果



(b) “浮雕”样式效果



(c) “渐变叠加”样式效果

图 5.79　对文字图层添加图层样式

5.4.2　滤镜

滤镜通过改变图像的像素数据，实现图像特殊处理效果。Photoshop 提供的内置滤镜主要分类放置在滤镜菜单中，使用时操作简单，只需从该菜单中选择相应的滤镜，设置相关参数，即可实现丰富的图像特殊效果。Photoshop 也允许使用第三方厂商生产的滤镜，称为外挂滤镜。外挂滤镜品种繁多，功能强大，能够满足设计人员更多方面需求。

使用滤镜时需要注意：如果仅是针对图像局部应用滤镜，而不是图像整体，则需要使用选区工具设定选取范围。

Photoshop CS3 提供了以下 14 组滤镜，每组又包括多个滤镜。

风格化：包含 9 种滤镜，通过替换像素、增加相邻像素对比度，产生夸张的效果。

画笔描边：包含 8 种滤镜，其中一部分滤镜产生绘画效果，另外滤镜则可以添加颗粒、绘画、边缘细节或纹理。

模糊：包含 11 种滤镜，通过降低相邻像素的对比度使图像产生模糊效果。

扭曲：包含 13 种滤镜，可以对图像进行几何扭曲、创建三维或其他变形效果。

锐化：包含 5 种滤镜，通过增强相邻像素的对比度使模糊的图像变得清晰。

视频：包含 2 种滤镜，将普通图像转换成视频设备可以接受的图像。

素描：包含 14 种滤镜，可以用来模拟素描和速写等艺术效果或手绘外观。

纹理：包含 6 种滤镜，可以给图像添加龟裂、马赛克、颗粒等纹理图案。

像素化：包含 7 种滤镜，将图像转换成带有色块的区域，从而产生晶格状、碎片、马赛克等效果。

渲染：包含 5 种滤镜，可以用来创建纤维、云彩、光照以及镜头光晕等效果。

艺术效果：包含 15 种滤镜，模拟彩色铅笔、蜡笔、油画以及木刻等作品特殊效果。

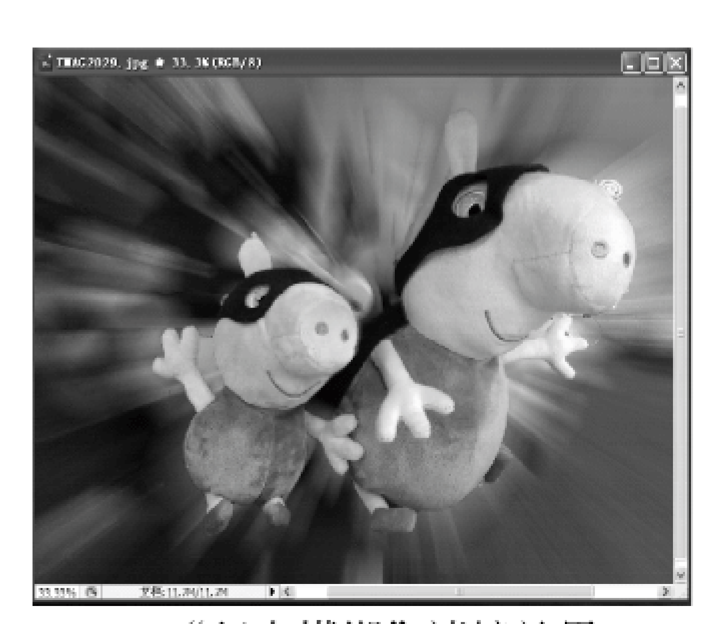
杂色：包含 5 种滤镜，杂色是随机分布的彩色像素点，通过在图像上添加或删除杂色除去图像上的瑕疵。

其他：包含 5 种滤镜，可以让用户创建自己的滤镜，使用滤镜修改蒙版或者在图像中使选区发生位移并快速调整颜色。

Digimarc：包含 2 种滤镜，可以将数字水印嵌入到图像中以存储版权信息。

除此之外，Photoshop CS3 还提供了 4 种功能强大的工具滤镜：抽出滤镜（用于抠图）、液化滤镜（用于各种变形）、图案生成器滤镜（用于制作各种图案）和消失点滤镜（用于构建平面的空间模型）。

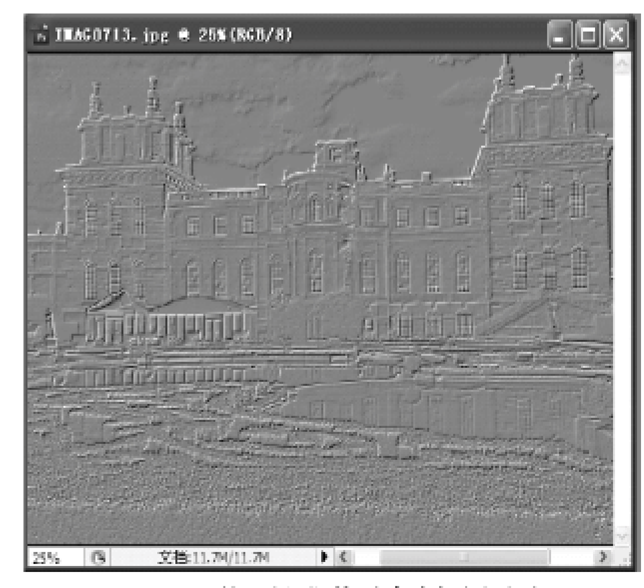
图 5.80 给出分别使用“径向模糊”、“水波扭曲”、“浮雕”滤镜制作的效果。



(a) “径向模糊”滤镜效果



(b) “水波扭曲”滤镜效果



(c) “浮雕”滤镜效果

图 5.80 使用滤镜处理图像

5.5 图像处理实例

5.5.1 制作网页导航菜单

网站的导航菜单主要用于引导用户打开相关页面，是网站构建不可或缺的元素。设计精美的导航菜单给人以赏心悦目的效果，网页设计人员通常会使用专业的设计软件进行导航菜单的设计。下面给出使用 Photoshop 制作导航菜单的详细步骤。

（1）启动 Photoshop，选择菜单“文件”→“新建…”，在弹出的“新建”对话框中设置宽度为 800 像素，高度为 22 像素，背景内容设为透明，单击“确定”按钮，如图 5.81 所示。



图 5.81　创建新文件

(2) 在工具箱中选择油漆桶工具 ，手动设置前景色的 R、G、B 值分别为 187、224、227，使用前景色填充图像窗口，如图 5.82 所示。





图 5.82　填充图像窗口

(3) 在工具箱中选择矩形工具 ，在属性栏单击“颜色”方框图标，在弹出的对话框中设置填充颜色的 R、G、B 值分别为 0、204、153，接下来在图像窗口绘制矩形区域。然后将图层面板切换到路径面板，单击面板下方的 图标，将路径作为选区载入，如图 5.83 所示。

(4) 在图层面板中选中矩形区域所在图层，单击鼠标右键，在弹出菜单中选择“栅格化图层”。然后双击图层名称将其更改为“首页”，如图 5.84 所示。

(5) 在工具箱中选择横排文字工具 T ，创建文字图层。设置“字体”、“大小”和“颜色”为黑体、12、黑色，然后输入“首页”文字，并将其移到“首页”图层位置。重复本步骤，创建多个



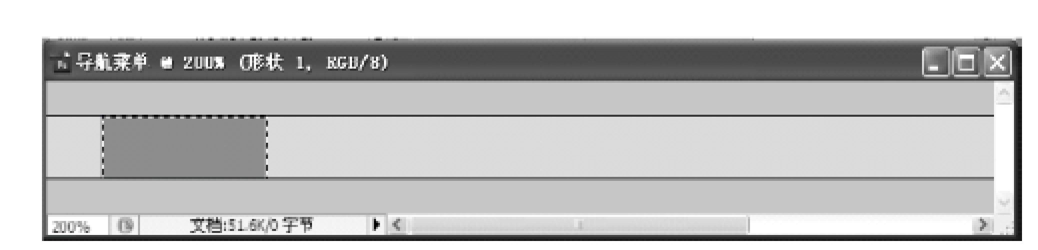


图 5.83　创建矩形填充区域

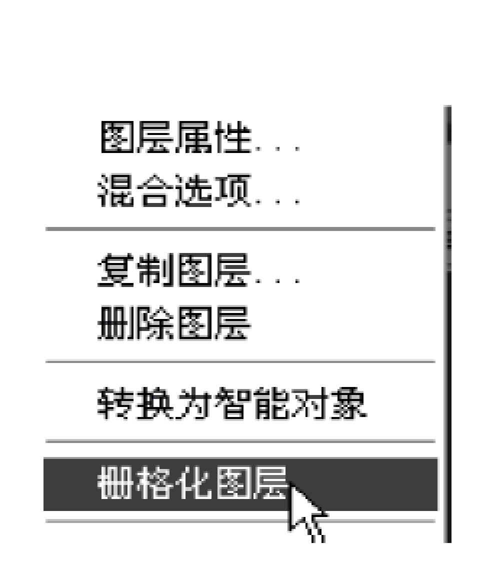




图 5.84　更改图层名称

文字图层,分别输入“站点地图”、“关于我们”、“登录|注册”文字,并移动到合适的位置。其中,“登录|注册”字体大小设置为 9,如图 5.85 所示。





图 5.85　创建文字图层

(6) 在工具箱中选择切片工具 ,对整个图像区域进行划分,如图 5.86 所示。

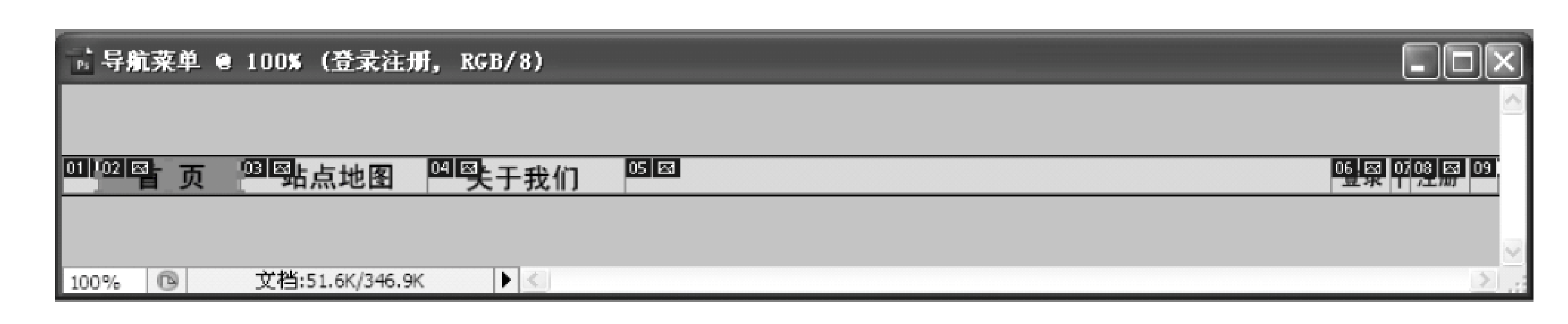


图 5.86　使用“切片工具”分割图像

(7) 选择菜单“文件”→“存储为 Web 和设备所用格式”,选择文件类型为 JPEG 格式,然后单击“存储”按钮,在打开的“将优化结果另存为”窗口中将保存类型设置为“仅图像”,最后单击“保存”按钮,如图 5.87 所示。

(8) 最终,导航菜单被分别导出成 JPEG 图像,如图 5.88 所示。

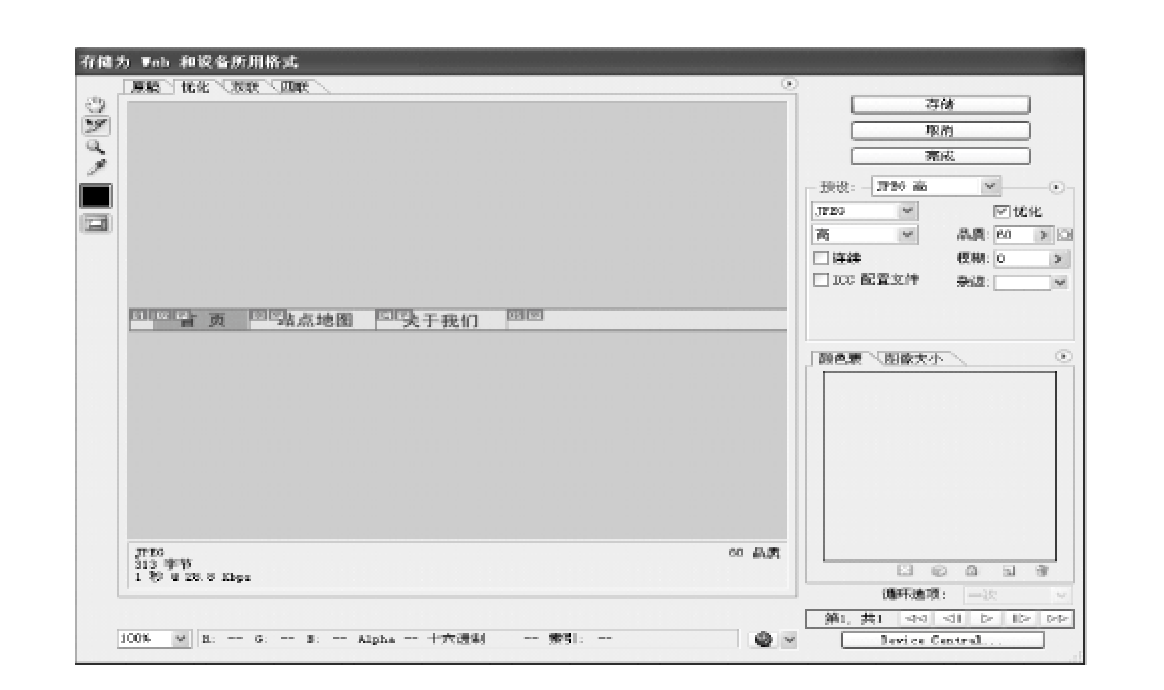



图 5.87　输出图像



图 5.88　JPEG 格式的导航菜单

5.5.2　美化网页图像

在网页中，添加图片会起到修饰网页、美化页面的效果。使用 Photoshop 可以帮助设计人员设计出精美的图片。下面举例说明如何使用 Photoshop 修饰、美化图片。

(1) 启动 Photoshop，选择菜单“文件”→“打开…”，打开“背景.jpg”文件。

(2) 在工具箱中选择横排文字工具 T ，设置“字体”、“大小”和“颜色”为黑体、31、白色，然后输入“咱们走着瞧—欢迎沃克健身俱乐部”文字，使用鼠标将文字图层移到合适位置。然后选择菜单“图层”→“栅格化”→“文字”，对文字图层执行“栅格化”操作，如图 5.89 所示。

(3) 在图层面板中选择文字所在的图层，单击图层面板下方的 fx. 图标，为该图层添加“斜面与浮雕”样式。在“图层样式”对话框中，设置“深度”为 62%，“方向”为“下”，“大小”为 1，“软化”为 0，“阴影角度”为－39 度，“阴影模式”为正常，“阴影颜色”为红色，“不透明度”为 100%，单击“确定”按钮，如图 5.90 所示。



图 5.89 添加文字

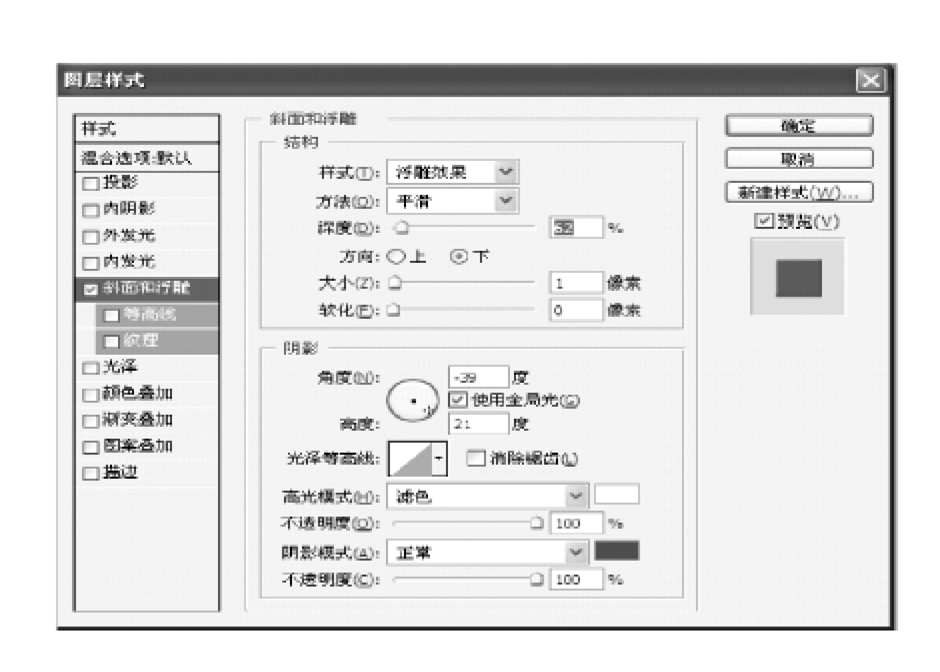




图 5.90 为文字添加“斜面与浮雕”样式

(4) 选择菜单“图层”→“新建”→“图层”,为该图层命名为“蓝天”。在工具箱中选择矩形选框工具 ,绘制矩形选区。然后在工具箱中选择吸管工具 ,在图像中蓝天区域单击鼠标左键,将蓝色设置为前景色,然后使用油漆桶工具填充矩形区域,如图 5.91 所示。

(5) 选择菜单“图层”→“新建”→“背景图层”,将背景图层转换成普通图层,并命名为“图层 0”,然后单击“蓝天”图层,将其拖动到“图层 0”的下方,如图 5.92 所示。



图 5.91 创建“蓝天”图层

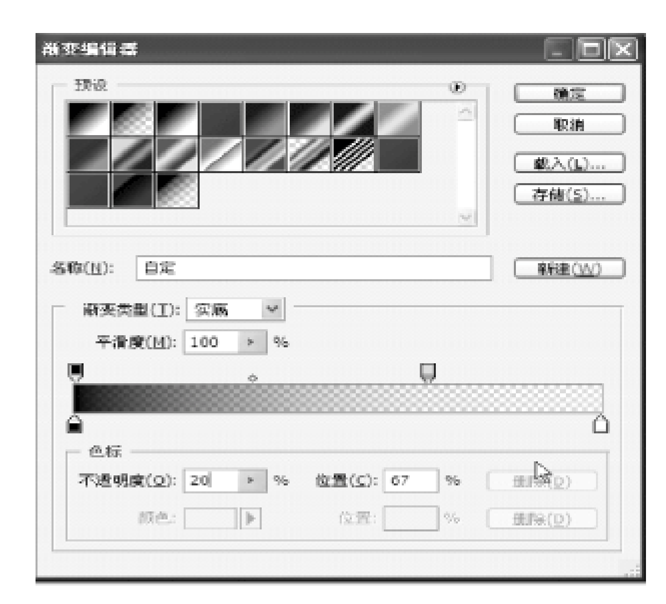
(6) 选中“图层 0”图层,然后选择菜单“图层”→“图层蒙版”→“显示全部”,为“图层 0”添加蒙版。在图层面板中,单击蒙版图标,此时蒙版图标外侧出现矩形框,表示已经选中蒙



图 5.92 调整“蓝天”图层次序

版。在工具箱中选择渐变工具，在“渐变编辑器”中选择“黑白”渐变，在下方颜色条中，设置左边的不透明色标的不透明度为100%，右边的不透明色标的“不透明度”为20%，“位置”为67%，然后在图像窗口中的蓝天区域进行线性填充，如图5.93所示。

（7）选择菜单“图层”→“拼合图层”，将图层面板中的图层拼合成一个图层，然后选择菜单“图像”→“调整”→“亮度/对比度”，在“亮度/对比度”对话框中设置“亮度”为30，如图5.94所示。



(a) “渐变编辑器” 对话框



(b) 选择蒙版



(c) 创建“蓝天”效果

图 5.93 使用蒙版创建“蓝天”效果



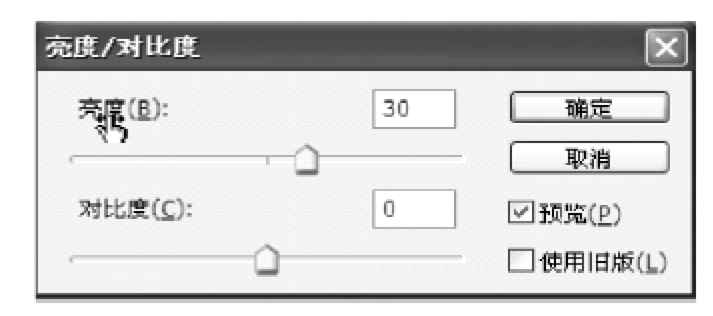


图 5.94 拼合图层并设置亮度

（8）选择菜单“文件”→“存储为”，将其保存为“banner.jpg”，最终效果如图5.95所示。





图 5.95 最终效果

5.5.3 制作网页效果图

在进行网站设计时,需要反复和客户进行界面交流,这时可以使用Photoshop制作出网页效果。网页效果图以直观的界面、美观的设计,让客户简略了解网站功能和界面风格。下面举例说明如何使用Photoshop制作网页效果。

(1) 启动Photoshop,选择菜单“文件”→“存储为”,新建1000×600像素文件,单击“确定”按钮,如图5.96所示。



图5.96 新建文件

(2) 选择菜单“视图”→“标尺”,在图像窗口四周显示标尺信息,方便精确作图。在标尺位置单击鼠标右键,弹出菜单,设置标尺单位为“像素”,如图5.97所示。

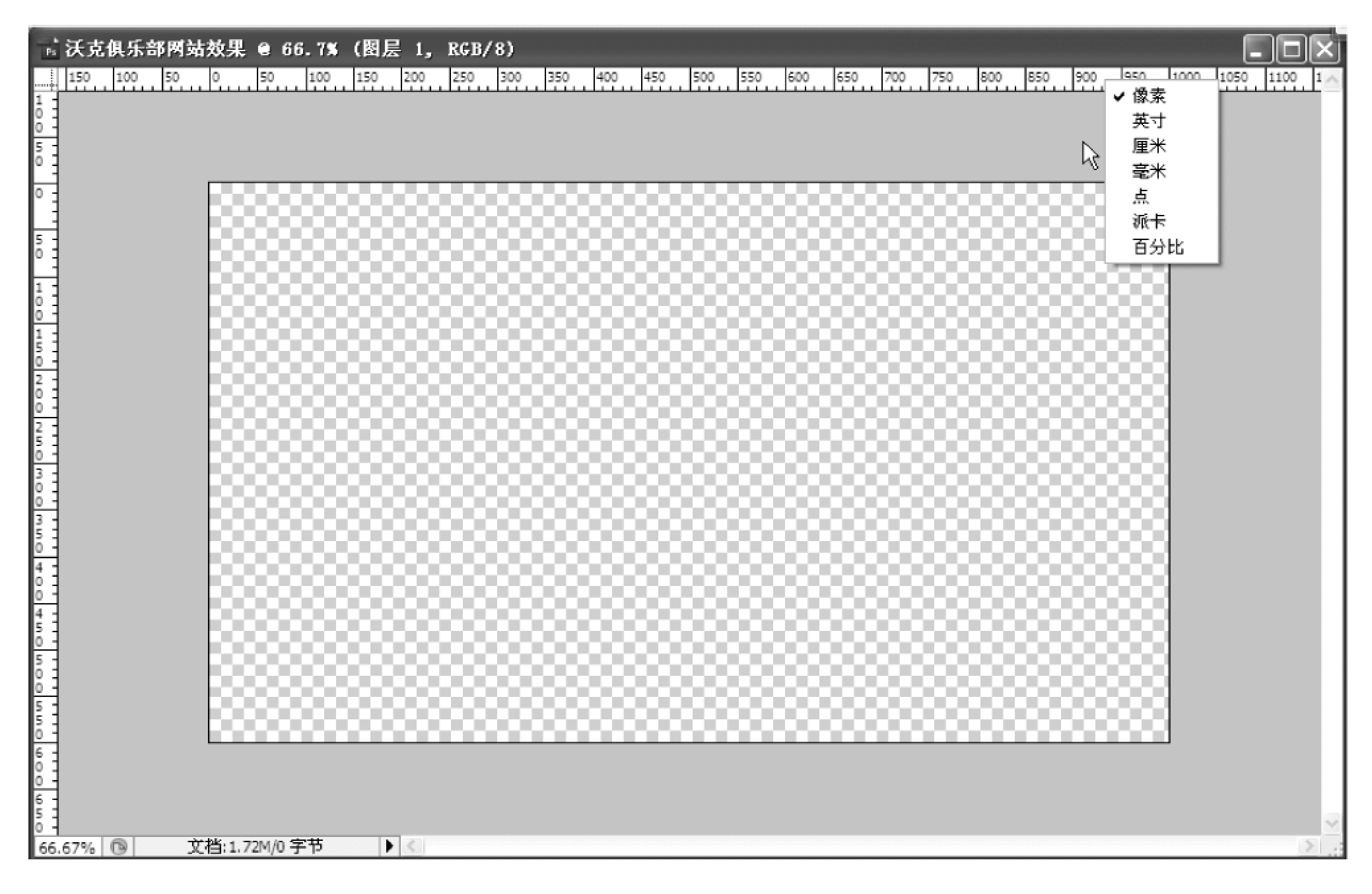


图5.97 设置标尺单位

(3) 在工具箱中选择油漆桶工具，设置前景色 R、G、B 值分别为 0、204、153，然后填充画布，如图 5.98 所示。

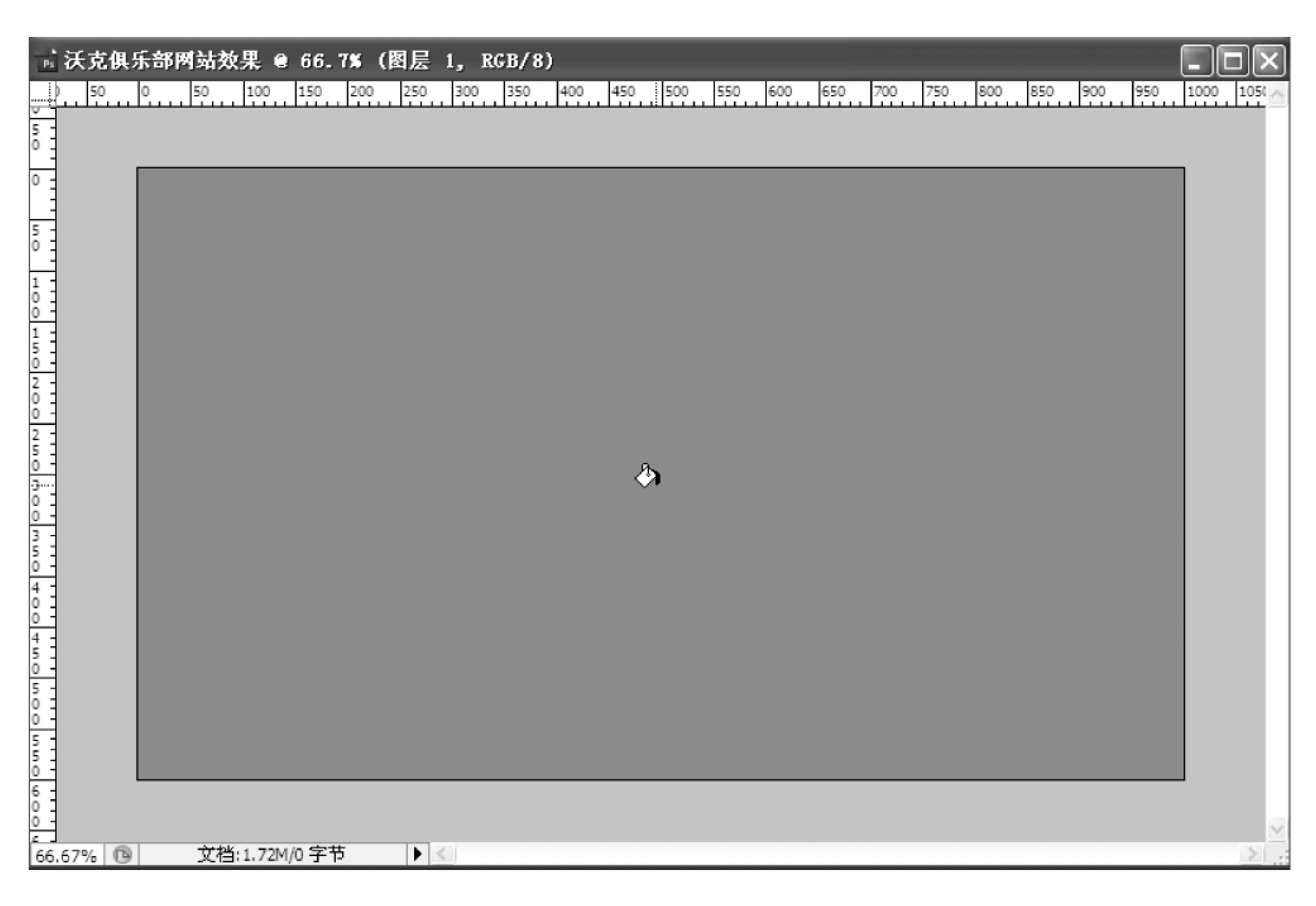


图 5.98 使用填充工具填充

(4) 选择菜单"文件"→"打开"，打开在 5.5.2 小节中制作的 banner.jpg 图像，选中"背景"图层，单击鼠标右键，在弹出菜单中单击"复制图层"，设置"复制图层"对话框中的目标文件为"沃克俱乐部网站效果"，将当前图层复制到"沃克俱乐部网站效果"图像中，如图 5.99 所示。





图 5.99 复制 banner.jpg 图像到"沃克俱乐部网站效果"图像中

(5) 切换到"沃克俱乐部网站效果"图像窗口中，双击复制的图层名称将其重命名为 banner，并使用工具箱中的移动工具将其移动到合适的位置，如图 5.100 所示。

(6) 重复执行步骤(4)和(5)，将 5.5.1 小节中制作的所有导航菜单图像都复制到"沃克俱乐部网站效果"图像文件中，然后对图层分别重命名为"空白导航菜单 1"、"首页"、"站点地图"、"关于我们"、"空白导航菜单 2"、"登录"、"|"、"注册"、"空白导航菜单 3"，并将其调整到合适位置，如图 5.101 所示。

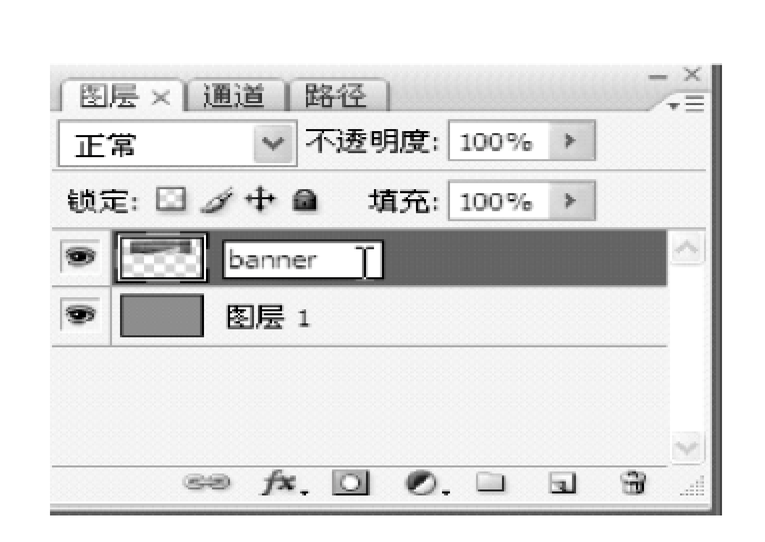




图 5.100 移动 banner 图层到合适位置





图 5.101 复制导航菜单图像到"沃克俱乐部网站效果"图像中

（7）选择菜单"图层"→"新建"→"图层"，命名为 container。选中该图层，选择工具箱中的矩形选框工具 ，创建 800×250 像素的选区，使用油漆桶工具 对其填充，填充颜色 R、G、B 值为 187、224、227，按住 Ctrl＋D 键取消选区，如图 5.102 所示。



图 5.102 创建 container 图层

(8) 选择工具箱中的横排文字工具 T ,设置字体、大小和颜色为黑体、14、黑色,然后输入文字"活动交流",选择移动工具 将其移动合适位置。重复本步骤,创建多个文字图层,分别输入文字"技术 & 装备"、"休闲驿站"和"网站事务",并调整到合适位置,如图 5.103 所示。





图 5.103 创建文字图层

(9) 重复步骤(7),新建 800×70 像素的图层,命名为 footer,使用油漆桶工具进行填充,填充颜色 R、G、B 值为 187、224、227,如图 5.104 所示。

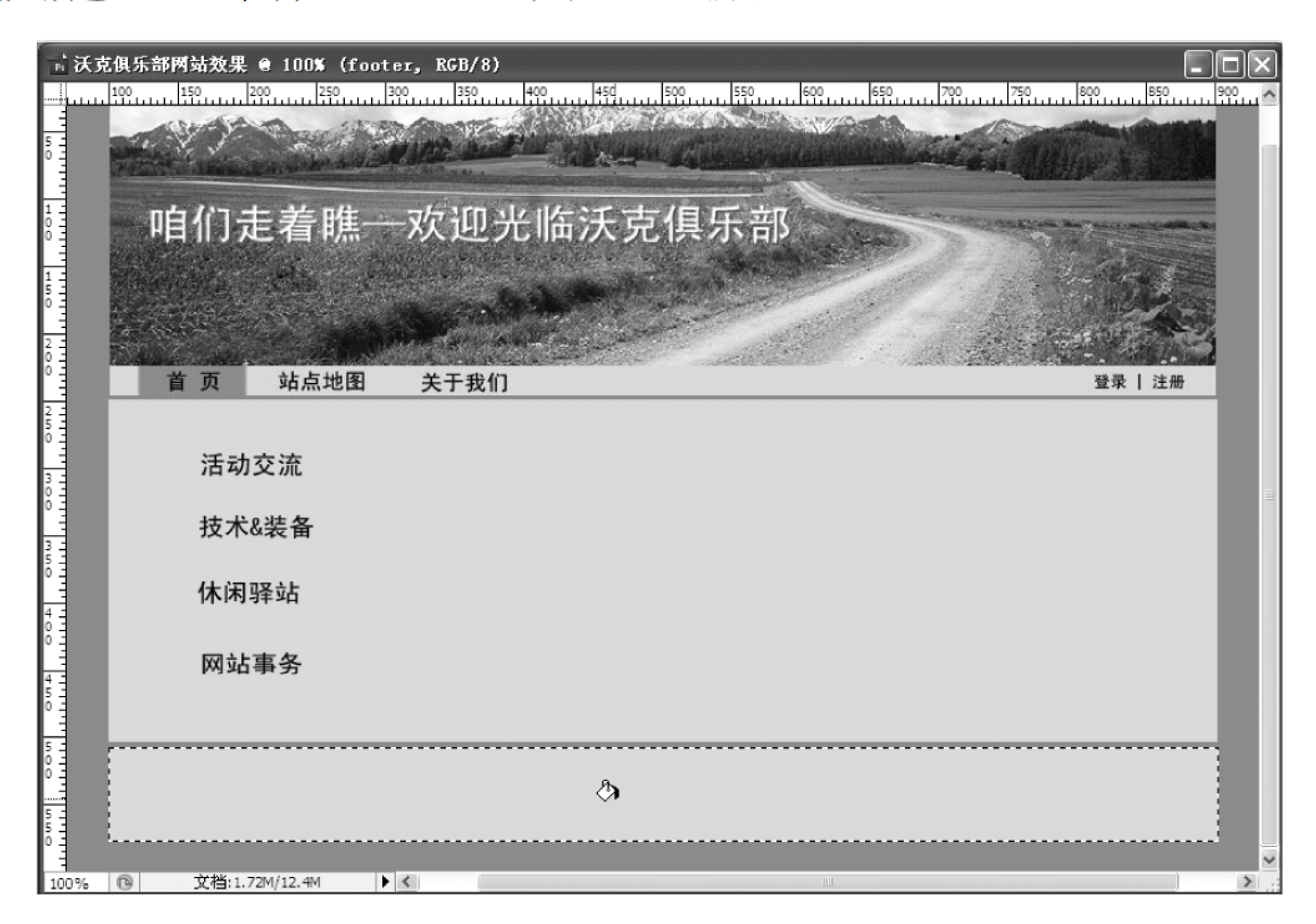


图 5.104 创建 footer 图层

(10) 选择工具箱中的横排文字工具 T ,创建文字图层。设置字体、大小和颜色为黑体、10、黑色,然后输入文字"联系我们| Email: webmaster@walkerclub.net"和"管理员入口",选择移动工具 将其移动到合适位置。

(11) 选择菜单“文件”|“存储为”，将其保存为“沃克健走俱乐部效果.jpg”。网站最终效果如图 5.105 所示。



图 5.105　最终网站效果图

小结

图像在网页中起到美化和点睛的作用，掌握 Photoshop 的图像处理方法对设计网页、美化网页界面起到重要作用。

本章首先介绍了 Photoshop 工具箱中各种工具的基本功能，以及如何应用这些工具创建选区、绘制及修饰图像，掌握这些工具是进行图像处理的基础。本章还介绍了 Photoshop 中图像基本操作和利用图层进行图像处理的基本知识，对这些知识的理解为图像进一步处理奠定了理论基础。本章还对滤镜知识进行了简单的介绍，使用滤镜能获得对图像处理的特殊效果。最后，为了更好理解和掌握网页图像处理过程，本章以实例为基础详细介绍了网页导航菜单、网页图像美化以及制作网页效果图的操作步骤。

通过本章的学习，读者能初步掌握网页图像处理的基础知识，对网页图像的高级处理仍需读者查阅更多资料完成。

习题

1. Photoshop 是________公司出品的用于进行________处理的工具。
2. 历史记录画笔工具主要用于进行________处理的工具。

3. 对图像进行渐变色填充应该使用________工具完成。
4. 对选区羽化的主要目的是______________________。
5. 容差是________工具用来创建选区需要设置的参数。
   A. 矩形选框工具　　B. 套索工具　　C. 磁性套索工具　　D. 魔棒工具
6. 制作图像的镜像效果,可以选择下面________变换实现。
   A. 斜切　　B. 水平翻转　　C. 垂直翻转　　D. 缩放
7. 使图像颜色更鲜艳,需要对图像的________进行处理。
   A. 明度　　B. 饱和度　　C. 亮度　　D. 色相
8. 采用________工具可以修复有瑕疵的图像。
   A. 图案图章工具　　B. 画笔修复工具　　C. 加深工具　　D. 油漆桶工具
9. 任选两个图像,使用图层蒙版将其合成一个图像,存储为 JPEG 格式。
10. 自己设计一个网页效果图。

第6章 网页制作工具Dreamweaver

6.1 初识 Dreamweaver

Adobe Dreamweaver CS5 是一款集网页制作和管理网站于一身的所见即所得网页编辑器,Dreamweaver CS5 是针对专业网页设计师特别发展的视觉化网页开发工具,利用它可以轻而易举地制作出跨越平台限制和跨越浏览器限制的充满动感的网页。

6.1.1 Dreamweaver 工作流程概述

用户可以使用很多种方法来创建 Web 站点,图 6.1 所示是一般网站制作的流程。

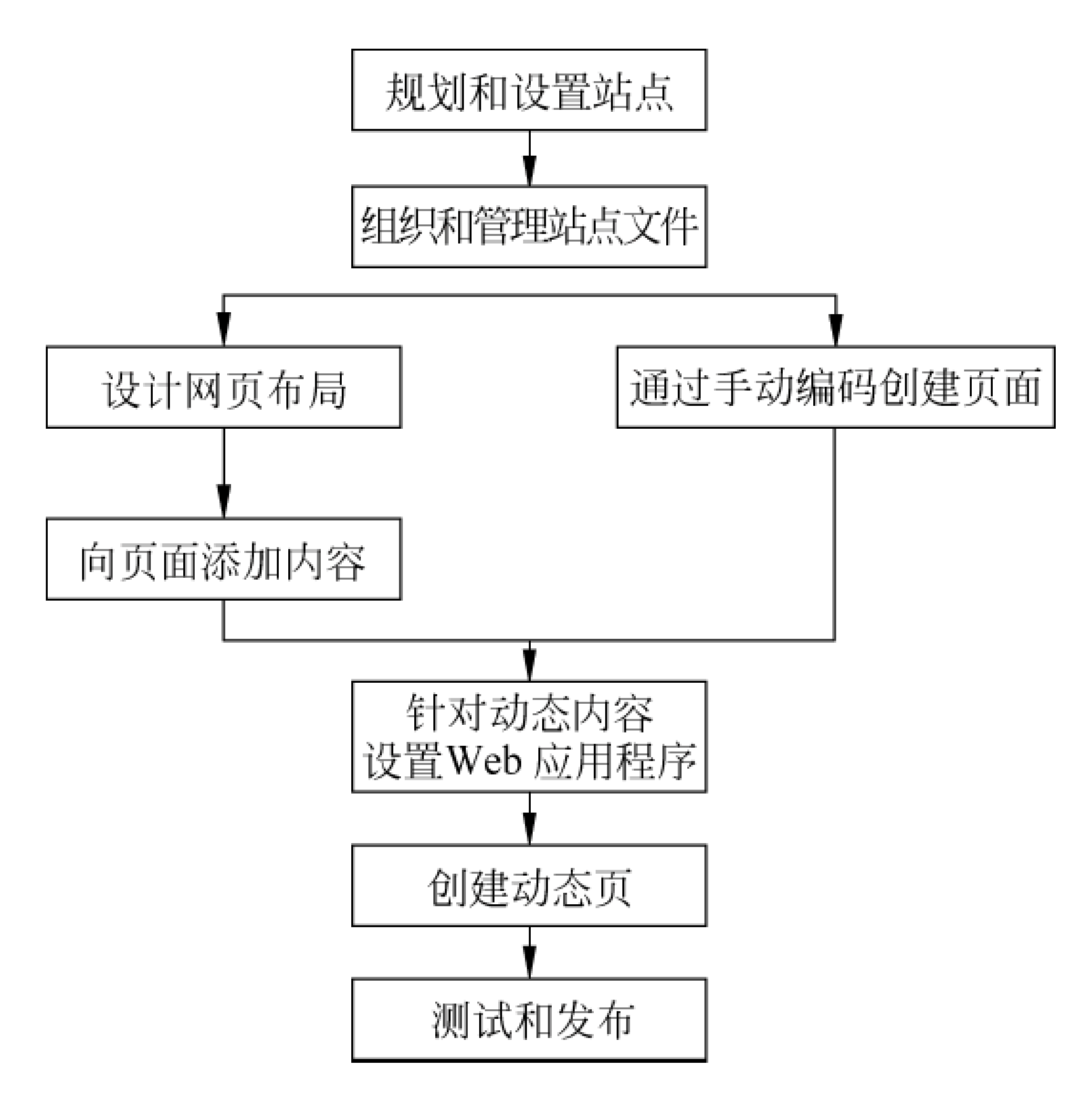


图 6.1 一般网站制作的流程

1. 规划和设置站点

确定将在哪里发布文件,检查站点要求,访问者情况以及站点目标。此外,还应考虑诸如用户访问以及浏览器、插件和下载限制等技术要求。在组织好信息并确定结构后,就可以开始创建站点。

2. 组织和管理站点文件

借助 Dreamweaver 提供的站点管理功能，根据需要更改组织结构。

3. 设计网页布局或者通过手动编码的方法创建页面

选择要使用的布局方法，或综合使用 Dreamweaver 布局选项创建站点的外观。设计完页面后要向页面添加内容，如文本、图像、鼠标经过图像、图像地图、颜色、影片、声音、HTML 链接、跳转菜单等。手动编写页面的代码是创建页面的另一种方法。Dreamweaver 提供了易于使用的可视化编辑工具，同时也提供了高级的编码环境，可以采用任一种方法(或同时采用这两种方法)来创建和编辑页面。

4. 针对动态内容设置 Web 应用程序

许多 Web 站点都包含了动态页，动态页使访问者能够查看存储在数据库中的信息，并且一般会允许某些访问者在数据库中添加新信息或编辑信息。若要创建动态页，必须先设置 Web 服务器和应用程序服务器，创建或修改 Dreamweaver 站点，然后连接到数据库。

5. 创建动态页

在 Dreamweaver 中，可以定义动态内容的多种来源，其中包括从数据库提取的记录集、表单参数和 JavaBeans 组件。若要在页面上添加动态内容，只需将该内容拖动到页面上即可。

6. 测试和发布

测试页面是在整个开发周期中进行的一个持续的过程。在这一工作流程的最后，在服务器上发布该站点。许多开发人员还会安排定期的维护，以确保站点保持最新并且工作正常。

6.1.2 认识 Dreamweaver CS5 的工作界面

启动 Dreamweaver CS5 将看到欢迎屏幕，如图 6.2 所示。

在“新建”栏中选择一个选项(如选择 HTML 选项)后即可进入工作界面，同时创建一个名字为 Untitled-1 的文件，如图 6.3 所示。

Dreamweaver CS5 的工作界面可以定制，图 6.3 工作界面比较简洁，对于一些常用的功能可以添加到工作界面，以方便使用。例如，打开“窗口”菜单，勾选“插入”即可在工作界面上添加“插入”工具栏，同样的方法可以在工作区放置其他功能组件。图 6.4 就是一个放置了常用功能组件的工作界面。

图 6.4 中的常用的元素功能如下。

1. 插入栏

“插入”栏可方便用户在制作网页时快速插入需要的网页元素。在 Dreamweaver CS5 中提供了“常用”、“布局”、“表单”、“数据”、Spry、InContext Editing、“文本”及“收藏夹”等 8

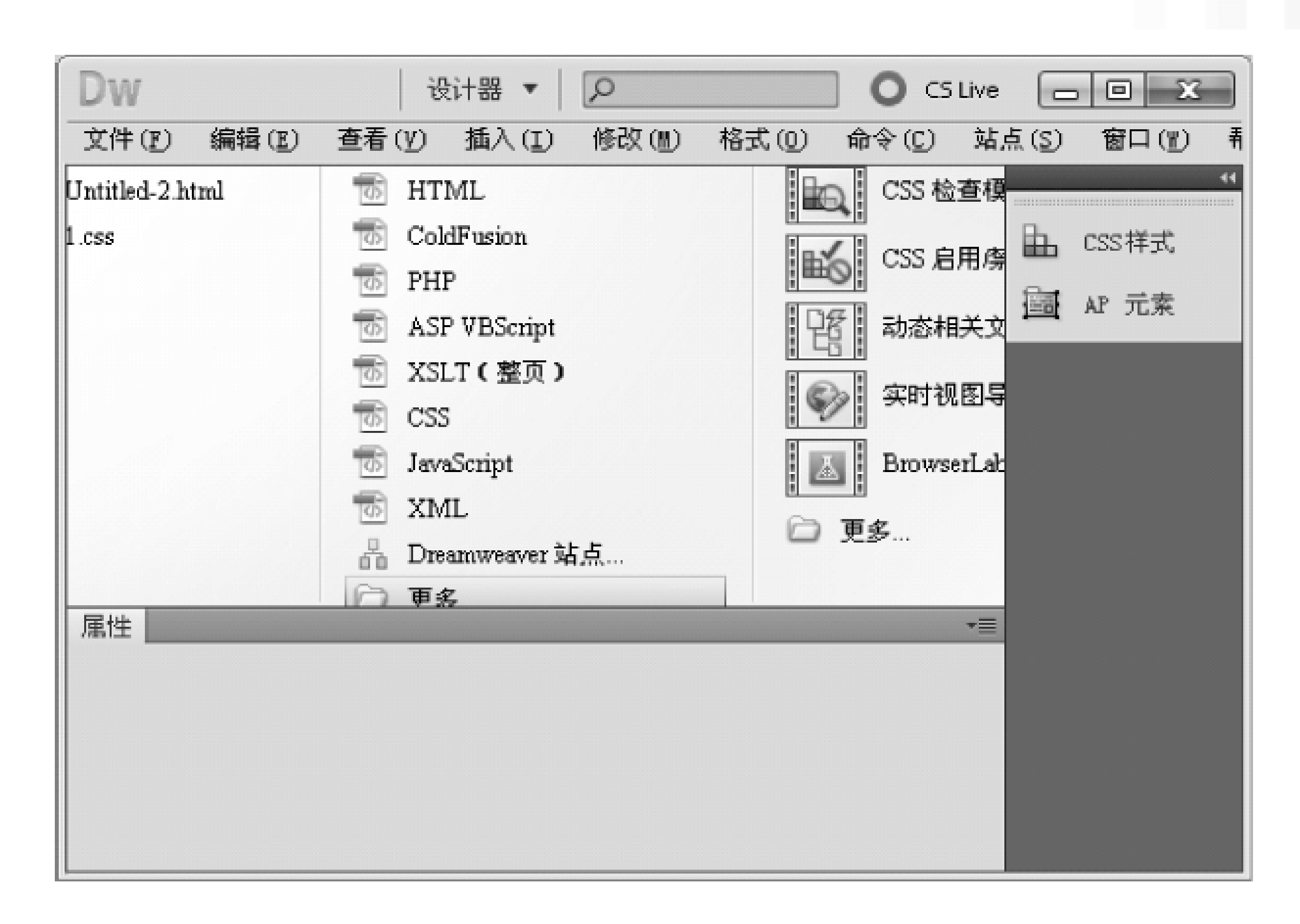


图 6.2　Dreamweaver CS5 欢迎画面

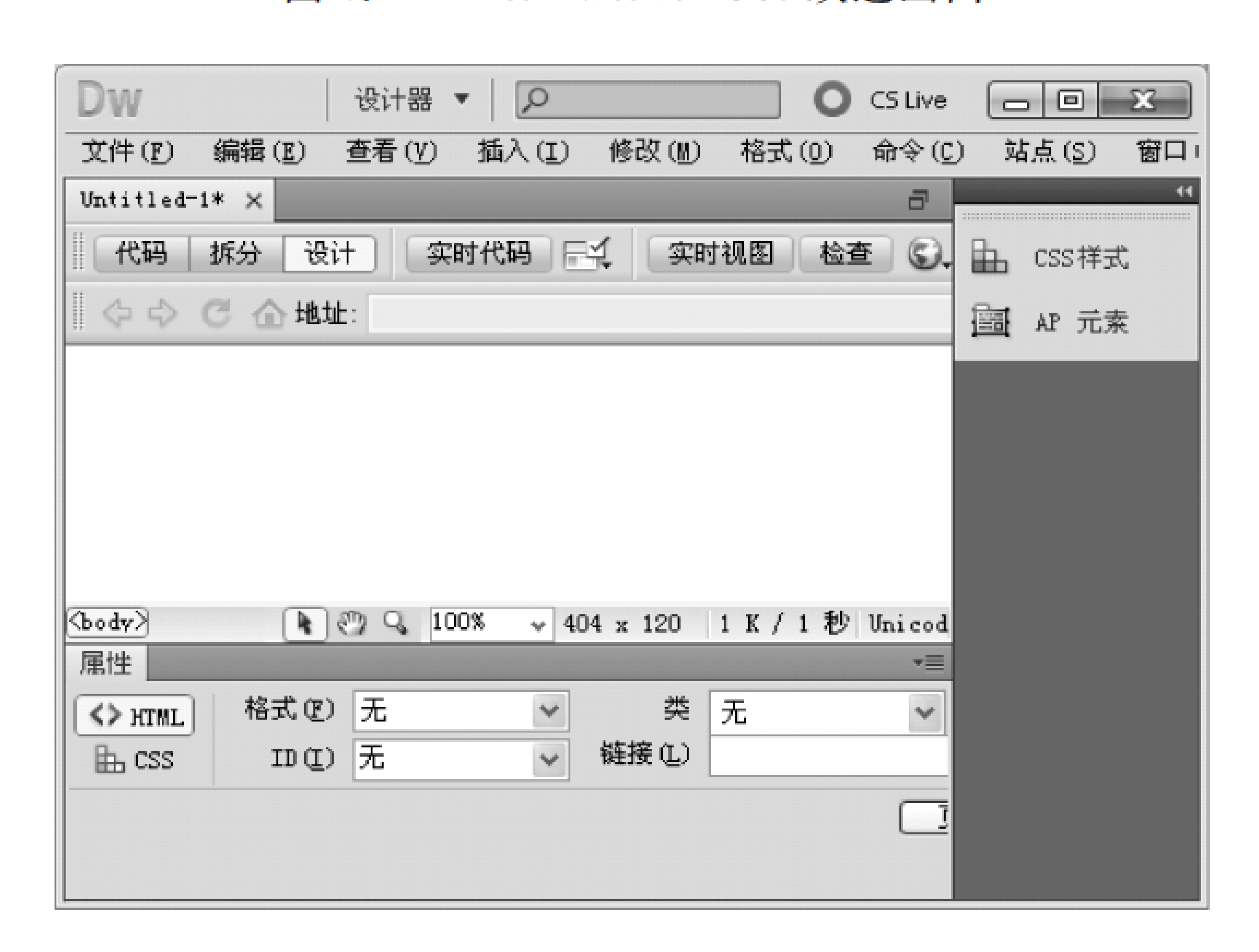


图 6.3　Dreamweaver CS5 的工作界面



图 6.4　添加了常用窗口的 Dreamweaver CS5 的工作界面

个类别。选择相应的选项卡,即可在各个类别之间进行切换。

"插入"栏有菜单和制表符(Tab 页)方式两种显示方式。默认情况下显示的是以 Tab 页方式显示的插入栏。可以在制表符下在任意类别上右击选择"显示为菜单"切换到菜单形式。在菜单形式下,单击"常用"按钮旁的▼按钮,在弹出的快捷菜单中选择"显示为制表符"命令,可以以 Tab 页方式显示。

2. 文档工具栏

文档工具栏主要用于切换编辑区视图模式、设置网页标题、进行标签验证以及在浏览器中浏览网页等。

- 单击 代码 按钮可切换到代码视图,其中显示了网页的源代码,适用于手动添加代码的情况。
- 单击 拆分 按钮则会在"文档"左边显示代码视图,在右边显示设计视图窗口,适用于既要看设置后的效果,又方便手动修改或添加代码的情况。
- 单击 设计 按钮则可以切换到设计视图,在其中可以以可视化的方式进行网页的编辑,是最常用的视方式。
- 单击 实时代码 按钮可以在"文档"窗口中显示当前打开的页面的代码。
- 单击 实时视图 按钮可以在"文档"窗口中显示当前打开的页面的预览。
- 单击 检查 按钮可以打开实时视图和检查模式。
- 单击 可以选择打开网页的浏览器。

3. 面板组

面板组一般置于编辑窗口右侧,是站点管理、事件添加等操作的场所。将鼠标光标移动到浮动面板的图标上,拖动该面板即可将浮动面板移动到窗口中的任何位置。

浮动面板组中的许多面板项目是非常有用的,可以通过选择"窗口"菜单中的相应菜单命令,显示或关闭相应的面板。要显示"文件"面板,可通过选择"窗口"→"文件"命令来实现。若想关闭某面板组,当面板展开时单击该面板组右上角的按钮,在弹出的菜单中选择"关闭标签组"命令即可。

4. 属性面板

"属性"面板通常用于设置和查看所选对象的各种属性,它位于 Dreamweaver CS5 窗口底部。单击其右上角的按钮可以关闭"属性"面板,也可以在"窗口"菜单中选择显示"属性"面板。不同对象其"属性"面板的参数设置项目也不同。如图 6.5 所示为"单元格"属性面板,如图 6.6 所示为"图像"属性面板。



图 6.5 "单元格"属性面板

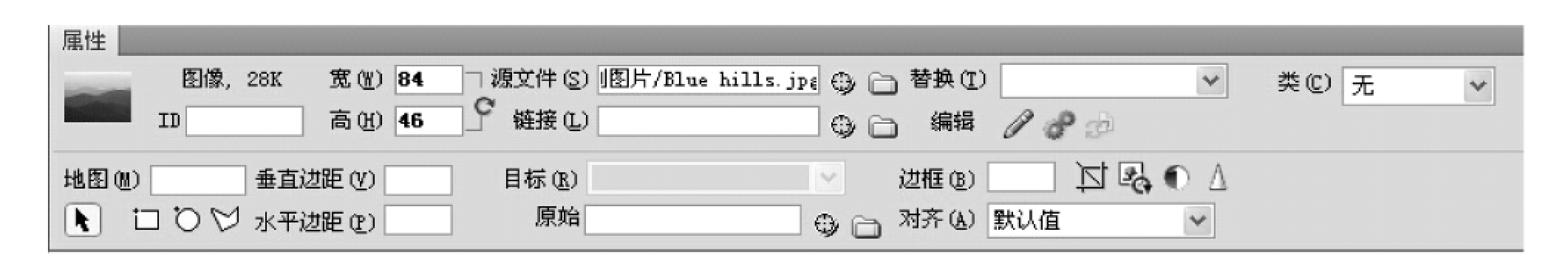


图 6.6　“图像”属性面板

5. 状态栏

状态栏用于显示标签、缩放文档大小、显示当前对象的大小等信息。其中包括标签选择器、选取工具、手形工具、缩放工具、设置缩放比率下拉列表框、窗口大小栏和文件大小栏等项目，如图 6.7 所示。



图 6.7　状态栏

标签选择器：在标签选择器中显示了一些常用的 HTML 标签，使用这些标签可以很方便地选择编辑区域中的某些项目，从而提高工作效率。例如要选择表格中的某一行，可以将光标定位到该行中的任意一个单元格中，然后单击标签选择器中的 <tr> 标签，如图 6.8 所示。

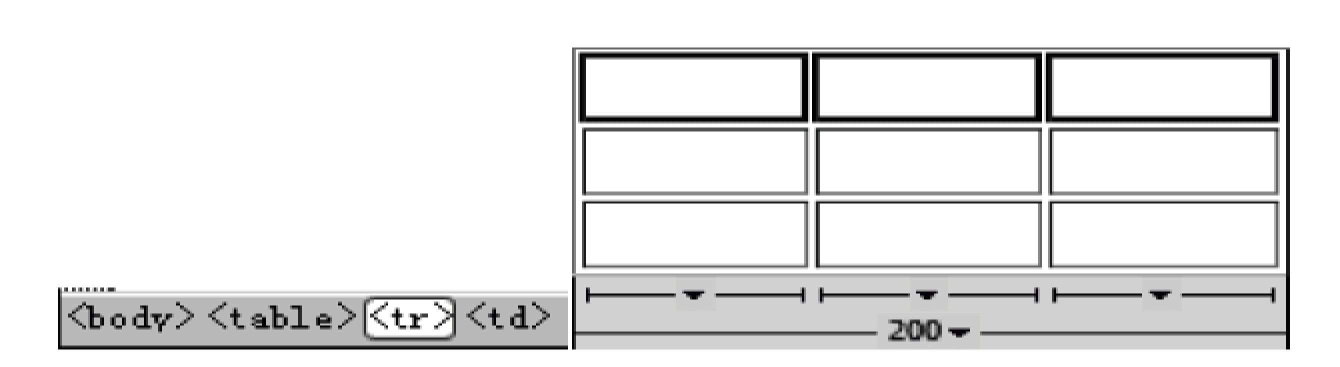


图 6.8　选择标签

- 选取工具 ：单击该工具，鼠标光标将变成箭头形状，此时可选择设计视图中的各种对象。
- 手形工具 ：单击该工具，鼠标光标将变成手形形状，此时按住鼠标左键不放并拖动鼠标，可移动设计视图中整个网页的位置，从而方便查看原来未显示出的部分内容。
- 缩放工具 ：单击该工具，鼠标光标将变成放大镜形状，此时在设计视图中单击鼠标左键可以放大显示设计视图中的内容；按住 Alt 键不放，鼠标指针将变为缩小镜形状，此时在设计视图中单击鼠标左键可以缩小显示设计视图中的内容；按住鼠标左键不放并拖动，在设计视图中脱出一个矩形框，释放鼠标，此时被矩形框框住的部分将以最大化的方式进行显示。
- 设置缩放比率下拉列表框：用于设置设计视图的缩放比率，可直接输入需要的缩放比率或单击右侧的 按钮，在弹出下拉列表中选择一个缩放比率选项。
- 窗口大小栏：用于显示当前设置视图的尺寸大小。
- 文件大小栏：用于显示当前网页文件的大小以及下载时需要的时间。

6.1.3 网页文档的基本操作

网页文档的基本操作包括创建网页、预览网页、保存网页、打开网页和关闭网页等，下面分别进行介绍。

1. 创建网页

在 Dreamweaver CS5 中可创建不同类型的空白网页文档，创建空白网页文档的操作步骤如下：

（1）启动 Dreamweaver CS5，选择“文件或新建”命令。

（2）在打开的“新建文档”对话框中选择“空白页”选项卡，在“页面类型”列表框中选择 HTML 选项，在“布局”列表框中选择“无”选项，如图 6.9 所示。

（3）完成空白网页文档的创建。



图 6.9 “新建文档”对话框

2. 预览网页

对网页进行编辑后，可以在预览器中对网页效果进行预览。选择“文件/在预览器中预览 IExplorer”命令或单击“文档”工具栏中的“在浏览器中预览调试”按钮，在弹出的快捷菜单中选择“预览在 IExplorer”命令即可执行预览网页的操作。

3. 保存网页

完成网页文档的编辑后，还需要对网页文档进行保存。

4. 打开网页

要对已有的网页进行修改，需要在 Dreamweaver 中打开网页，打开网页的主要方法有如下四种：

(1) 通过对话框打开：选择“文件”→“打开”命令，在打开的“打开”对话框的“查找范围”下拉列表框中选择网页所在的位置，在文件列表框中双击需要打开的网页。

(2) 通过菜单命令打开：选择“文件”→“打开最近的文件”命令，在弹出的子菜单中选择一个最近打开过的网页文件。

(3) 通过双击方式打开：对于后缀名为.asp、.aspx、.php、.jsp 等格式的网页文件，直接找到并双击该文件也可打开该网页。

(4) 通过快捷菜单打开：对于所有 Dreamweaver CS5 支持的网页文件，可以在该网页文件上单击鼠标右键，在弹出的快捷菜单中选择“使用 Adobe Dreamweaver CS5”命令。

6.1.4 设置页面属性

新建网页文档后，通常需要对网页文档的属性进行设置，主要包括网页的外观、超级连接样式和网页标题等。

要设置网页属性，可先打开网页文档，然后选择“修改”→“页面属性”命令，在打开的“页面属性”对话框左侧的“分类”列表框中选择类别，在右侧面板上进行各属性的详细设置。

1. 设置外观属性

打开“页面属性”对话框后，默认为外观类别，在其右侧可进行页面外观属性的详细设置。包括网页文档中文本字体、文本大小、文本颜色、网页文档背景图像等的设置。设置网页外观属性的操作步骤如下：

(1) 新建一个空白的网页文档，选择“修改”→“页面属性”命令，打开“页面属性”对话框。在“页面字体”下拉列表框中选择“新宋体”选项，在“大小”下拉列表框中选择“12”选项。

(2) 将文本颜色设置为黑色(＃0000000)，背景颜色设置为白色(＃FFFFFF)。

(3) 在“背景图像”文本框中输入背景图像的位置名称，在“重复”下拉列表框中选择“不重复”选项。

(4) 在“左边距”、“右边距”、“上边距”和“下边距”文本框中都输入“0”，激活其后的单位下拉列表框，保持默认设置，单击“确定”按钮完成页面外观属性的设置。

2. 设置链接属性

设置链接属性是指设置网页文档中超链接状态的格式，如超链接文本初始显示的颜色、当鼠标指向它时指定显示的颜色及单击它后显示的颜色等。设置网页文档超链接属性的操作步骤如下：

(1) 在打开的“网页属性”对话框的左侧“分类”列表框中选择“链接”选项，在“链接”下拉列表框右侧单击“加粗”按钮将链接文本设为加粗显示，设置字号为 9，在其后的下拉列表选框中选择“点数 pt”选项，如图 6.10(a)所示。

(2) 在“连接颜色”文本框中输入链接文本初始显示颜色，如蓝色(＃0033CC)；在“变换

图像链接”文本框中输入当鼠标指针位于链接文本上时显示的颜色，如淡蓝色(#00CCFF)，如图 6.10(b)所示。



(a) 设置链接文本的字体



(b) 设置链接文本的颜色

图 6.10 设置链接文本字体与颜色

(3) 在“已访问链接”文本框中输入当鼠标单击链接文本后显示的颜色，如紫色(#993399)；在“活动链接”文本框中输入当鼠标在链接文本上单击时显示的颜色，如橘黄色(#FF6600)，如图 6.11 所示。

(4) 在“下划线样式”下拉列表框中选择“仅在变换图像时显示下划线”选项，如图 6.12 所示。

3. 设置标题属性

在“页面属性”对话框的“分类”列表框中选择“标题”选项后，在其右侧面板中可设置网页中标题的字体、大小和颜色等参数，如图 6.13 所示。其设置方法和设置外观的样式基本相同。若选择“标题/编码”选项，对话框右侧面板中将显示“标题/编码”的参数设置项，如图 6.14 所示。在“标题”文本框输入网页标题，该标题将出现在浏览器标题栏中；在“文档



图 6.11　设置已访问链接文本的颜色



图 6.12　设置链接文本的下划线样式

图 6.13　设置网页中的标题样式

类型"下拉列表框中可以选择编辑的文档类型,通常选择 XHTML 1.0 Transitional 或 XHTML 1.0 Strict,可以使 HTML 文档与 XHTML 兼容;在"编码"下拉列表框中设置页面使用的字体编码类型。



图 6.14 0 设置网页在浏览器中的标题/编码

6.1.5 规划与创建站点

站点是管理网页文档的场所,利用站点的管理功能可对站点中的文件进行管理和测试。在创建站点之前还需先规划站点,以达到最佳效果。

1. 认识站点

通过各种链接联系起来的多个网页文档就构成了站点。Dreamweaver CS5 中提供了本地站点、远程站点和测试站点 3 类站点。

本地站点:是用户工作的目录,用于存放用户网页、素材等本地文件夹,在制作一般网页时只需建立本地站点即可。

远程站点:为了在不链接 Internet 的情况下而能对所建的站点进行测试、修改,需要在本地电脑上创建远程站点,以模拟真实的 Web 服务器环境进行测试。

测试站点:主要用于对动态页面进行测试,如在制作 ASP、PHP 或 JSP 等动态网页时必须创建测试站点,否则浏览网页时它将不能正确显示。

2. 规划站点

规划站点是指利用不同的文件夹将不同的网页内容分门别类地保存,合理地组织站点结构,可提高工作效率,加快对站点的设计。站点规划的目的在于明确建站的方向和确定实现这一目标所采用的方式,同时也是确定本站点所需要实现的功能。

规划站点时要明确网站的主题,哪些信息是必需的,哪些信息是希望有的,哪些信息是可以没有的,通过这些分析即可得出网站的栏目及其包含的内容。在规划站点结构时,最常采用的是树形模式的规划法。首先分频道,频道内容再次划分栏目,栏目内又划分出几个不同的子栏目,依次类推,形成树根式的模式图。

3. 创建并管理站点

站点规划完成后，即可开始创建站点。在 Dreamweaver CS5 中可以通过“站点管理”对话框进行站点的创建和管理操作。

(1) 创建站点

在一个网站中包含多个网页，为了更好地对这些网页进行管理和维护，在制作网页之前，最好先在 Dreamweaver CS5 中创建站点。

选择“站点/新建站点”命令，打开站点定义的对话框，在“站点名称”文本框中输入站点名称，这里输入“MySite”，如图 6.15 所示，单击“下一步”按钮，创建一个站点，如图 6.16 所示，新建的站点中没有文件，是空站点。



图 6.15　新建站点对话框

(2) 管理站点

创建站点完成后，还可以对站点进行管理。如添加测试服务器、改变站点的名称、创建文件或文件夹、删除不需要的文件或文件夹等。管理站点的操作如下：

① 选择“窗口/文件”命令，打开“文件”面板，在站点根目录上单击鼠标右键，在弹出的快捷菜单中选择“新建文件夹”命令，如图 6.17 所示。



图 6.16　站点视图



图 6.17　选择“新建文件夹”命令

② Dreamweaver将自动在站点根目录下创建一个名为untitled的新文件夹且文件夹名称处于可改写状态，修改文件夹的名称为相应栏目的名称，如about，然后单击Enter键确认，如图6.18所示。

③ 用相同的方法新建其他栏目的文件夹、子文件夹和图像文件夹。

④ 选择某一文件夹，在其上单击鼠标右键，在弹出的快捷菜单中选择“编辑/删除”命令，在打开的确认删除对话框中单击“是”按钮，将该文件夹从站点中删除。

⑤ 选择“站点/管理站点”命令，打开“管理站点”对话框，在站点列表中选择MySite站点，如图6.19所示。



图6.18 重命名文件夹



图6.19 “管理站点”对话框

⑥单击“编辑”按钮，打开如图6.15所示对话框中即可对该站点进行编辑，包括本地信息、远程信息、测试服务器以及站点地图布局等，其操作过程和前面创建站点的操作一样。

⑦ 编辑完成后，单击“保存”按钮关闭对话框，返回“管理站点”对话框，单击“删除”按钮，在打开的警告对话框中单击“是”按钮确认删除，再单击“管理站点”对话框中的“完成”按钮，删除MySite的站点，“文件”面板中将不再显示该站点。

6.2 制作简单网页

6.2.1 向网页中添加文本

文本是网页中运用最广泛的元素之一，它是向浏览者传递有效信息最主要的方式。文本是网页中不可或缺的元素，其信息传递的方式是其他任何一种网页元素都无法代替的。下面将详细讲解在网页中添加不同类型文本的方法。

1. 添加普通文本

在Dreamweaver CS5中通过“文本”插入栏可快速添加各类型的文本。当“插入”栏以菜单的方式显示时，单击“插入”栏的按钮，在弹出的菜单中选择“文本”命令即可切换到“文本”插入栏，如图6.20所示，鼠标移到相应的按钮上，会显示出按钮的名称。

在网页中添加文本与在Word等文字处理软件中添加文本一样方便，可以直接输入文

图 6.20　“文本”插入栏

本，也可从其他文档中复制文本，还可以导入其他文档中的文本。

方法一：直接输入文本

直接输入文本的方法很简单，只需将插入点定位在网页“文档”窗口中，切换到所需的输入法，然后直接输入所需的文本即可。

方法二：从其他文档中复制文本

从其他文档中复制文本可以节省时间，从而提高制作网页的速度，在其他文档中(如记事本、Word 文档等)选择需复制的文本。按 Ctrl+C 键或单击鼠标右键，在弹出的快捷菜单中选择“复制”命令将其复制到剪切板中，将插入点定位到网页中需输入文本的位置，按 Ctrl+V 键或单击鼠标右键，在弹出的快捷菜单中选择“粘贴”命令即可将剪切板中的文本粘贴到当前编辑窗口中。

方法三：导入文本

使用 Dreamweaver CS5 可以导入 XML 模板、表格数据、Word 及 Excel 等文档中的文本。

例 6.1　从 Word 文档导入文本。

(1) 新建一个空白网页文档，将插入点定位到“编辑”窗口中。

(2) 选择“文件”→“导入”→“Word 文档”命令，在打开的“导入 Word 文档”对话框中选择文档，如图 6.21 所示。



图 6.21　“导入 Word 文档”对话框

(3) 单击“打开”按钮，即可将文本内容导入到 Dreamweaver CS5 中。

(4) 保存文档命名为 Example6-1.html，如图 6.22 所示。

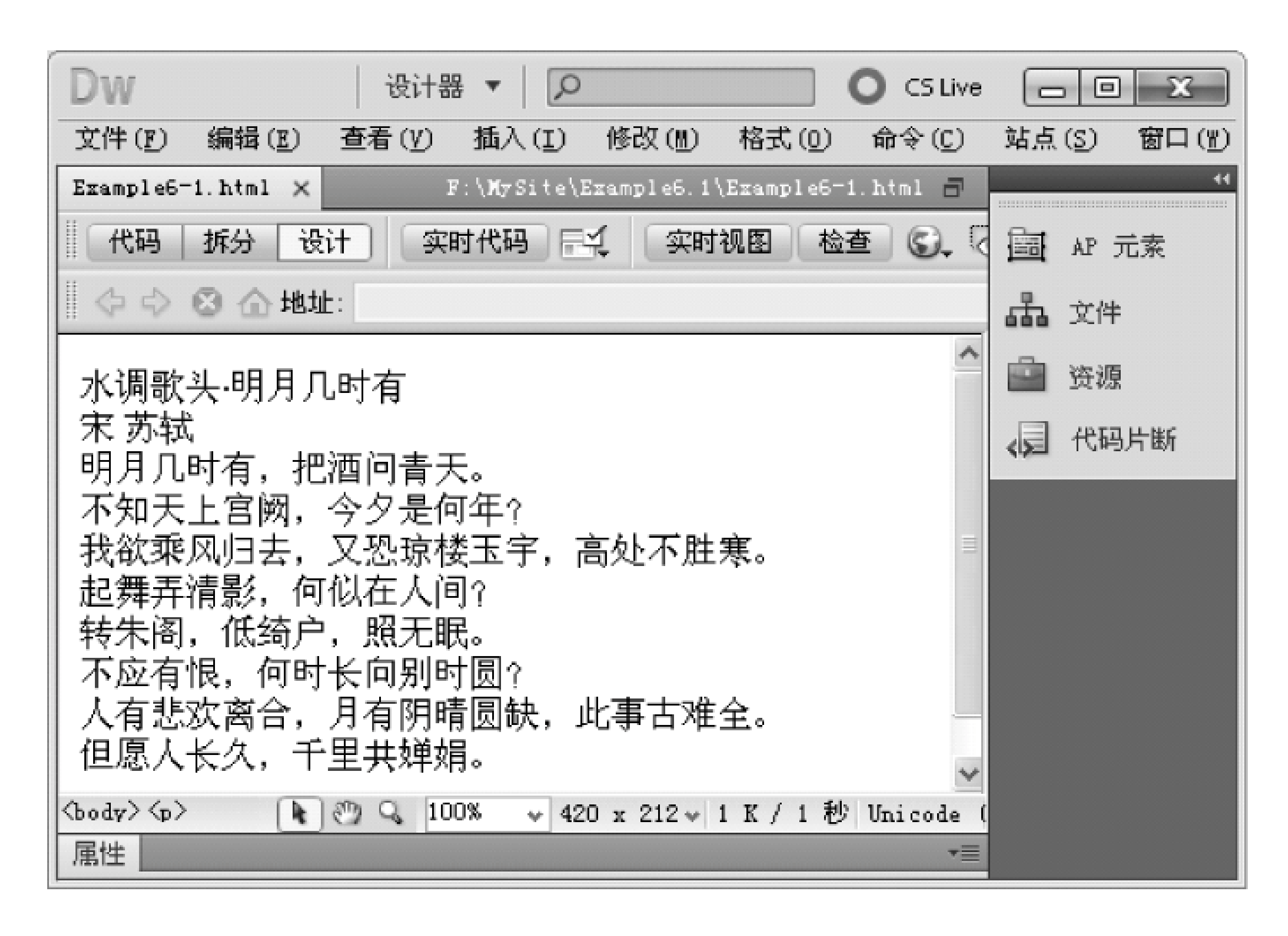


图 6.22　导入文本

2. 添加特殊字符



图 6.23　选择特殊符号

网页制作过程中，时常需要添加一些特殊字符，如版本符号©、注册商标符号®等。下面用不同的方法在网页中添加特殊符号，其操作步骤如下：

(1) 将插入点定位到“编辑”窗口中，将插入栏切换到“文本”插入栏。

(2) 单击“文本”插入栏中 按钮，在弹出的如图 6.23 所示的菜单中选择所需的命令即可添加常用的特殊符号。

(3) 若需添加更多的特殊字符，可在弹出的菜单中选择“其他字符”命令。

(4) 在打开的“插入其他字符”对话框中选择需要的字符，如图 6.24 所示。完成后单击“确定”按钮即可添加所需的字符。



图 6.24　“插入其他字符”对话框

3. 添加空格

在 Dreamweaver CS5 中输入空格时会发现无论按多少次空格键都只会出现一个空格，这是因为 Dreamweaver CS5 的文档格式都是以 HTML 形式存在，而 HTML 文档中只允许字符之间包含一个空格。要在网页文档中添加连续的空格，可以采用以下几种方式：

- 在“文本”插入栏中单击“已编排格式”按钮 PRE，再连续按空格键。
- 选择“插入”→HTML→“特殊字符”→“不换行空格”命令可以添加一个空格，如果需要添加多个空格可重复操作。
- 按 Shift＋Ctrl＋空格键添加一个空格，如果需要添加多个空格可重复操作。
- 将输入法切换到全角状态(通常按 Shift＋空格键可以进行全、半角状态切换)，直接按空格键，需要多少个空格就按多少次空格键。

4. 添加水平线

当网页中的对象较多时，页面可能会显得有些杂乱，这时可以使用水平线将多个对象进行分割，使段落区分更显明，让网页更具层次感。其操作步骤如下：

(1) 将光标定位到添加水平线后面。

(2) 选择“插入”→HTML→“水平线”命令添加水平线。

(3) 选择该水平线，在“属性”面板中取消选中的“阴影”复选框，在“高”文本框中输入“3”，在“宽”文本框里输入“100”，在其后的下拉列表框中选择“%”选项，在“对齐”下拉列表框中选择“居中对齐”选项，如图 6.25 所示。



图 6.25　水平线属性面板

(4) 单击“属性”面板右侧的快捷标签编辑器按钮 ，在“编辑标签”文本框中的“/>”文本前输入代码“color=/#FF0000”，如图 6.26 所示。

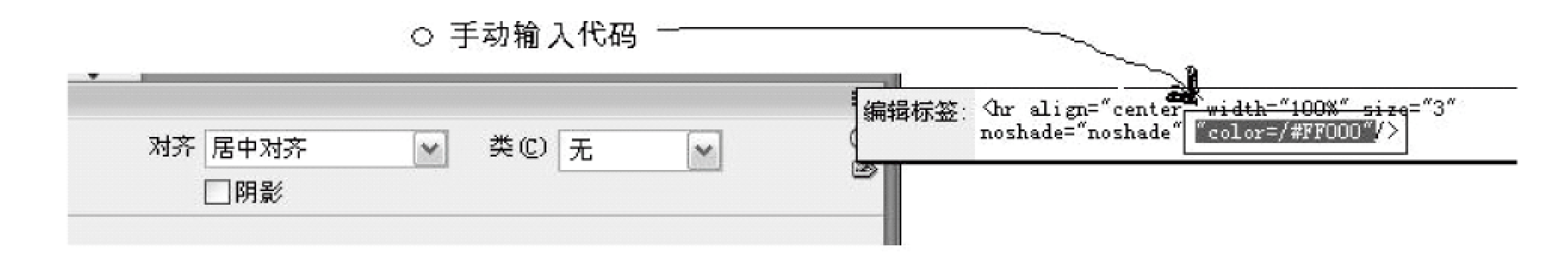


图 6.26　设置水平线的属性

(5) 输入完成后单击 Enter 键确认输入，保存页面，然后可按 F12 键浏览页面。

5. 添加日期

Dreamweaver CS5 提供了一个方便的日期对象，该对象可以以任何格式插入当前日期(包含或不包含时间)，并可在每次保存文件时都自动更新日期。

在 Dreamweaver CS5 中添加当前日期的操作步骤如下：

（1）将插入点定位到需要添加日期或时间的位置，选择“插入”→“日期”命令。

（2）在打开的“插入日期”对话框中设置星期、日期和时间格式等，若选中“储存时自动更新”复选框可在每次保存文档时都更新添加的日期。如果希望日期在插入后变成纯文本并永远不自动更新，请取消选择该选项，如图6.27所示。

（3）单击“确定”按钮完成日期的添加。



图6.27　网页中添加的日期和时间

6. 设置文本格式

合理设置网页文本的属性，可以使网页更美观、层次更分明。在“属性”面板中可对文本的颜色、字体和大小等进行设置，选择文本后的“属性”面板如图6.28所示。



图6.28　文本“属性”面板

7. 设置段落格式

设置网页文本的段落格式包括设置段落缩进和对齐方式。选择要设置段落格式的文本，或将插入点定位到该段落中，在“属性”面板中即可设置其段落格式。

6.2.2　向网页中添加图像

网页中的文本向人们传递了最翔实的信息，而网页中的图像却能起到画龙点睛的作用。恰到好处地在网页中应用图像，能使制作出的网页更加生动、美观。下面将介绍如何在网页中插入图像、编辑图像以及制作鼠标经过图像和导航条等。

1. 插入图像

网页中的图像一般有两种存在方式，一是作为网页的内容存在，二是作为网页或其他对象的背景存在。下面在Dreamweaver CS5中使用直接插入的方式为网页文档插入一幅图像作为网页内容，其操作步骤如下：

（1）新建空白文档，将插入点定位到需要插入图像的位置。

（2）选择“插入”→“图像”命令或将插入栏切换到“常用”插入栏，单击“图像”按钮，打开“选择图像源文件”对话框。

（3）在“查找范围”下拉列表框中选择要插入的图像的位置，然后在中间列表框中选择需插入到网页中的图像文件，“文件名”文本框中自动出现所选文件的名称，如图 6.29 所示。



图 6.29　选择图像

（4）单击“确定”按钮，在打开的“图像标签辅助功能属性”对话框的“替换文本”下拉列表框中，输入当用户浏览网页时图像不能正常显示或将鼠标光标移到图像上时显示的提示文本，在“详细说明”文本框中可指定对该图像进行详细说明的网页文档的路径及名称，如图 6.30 所示。

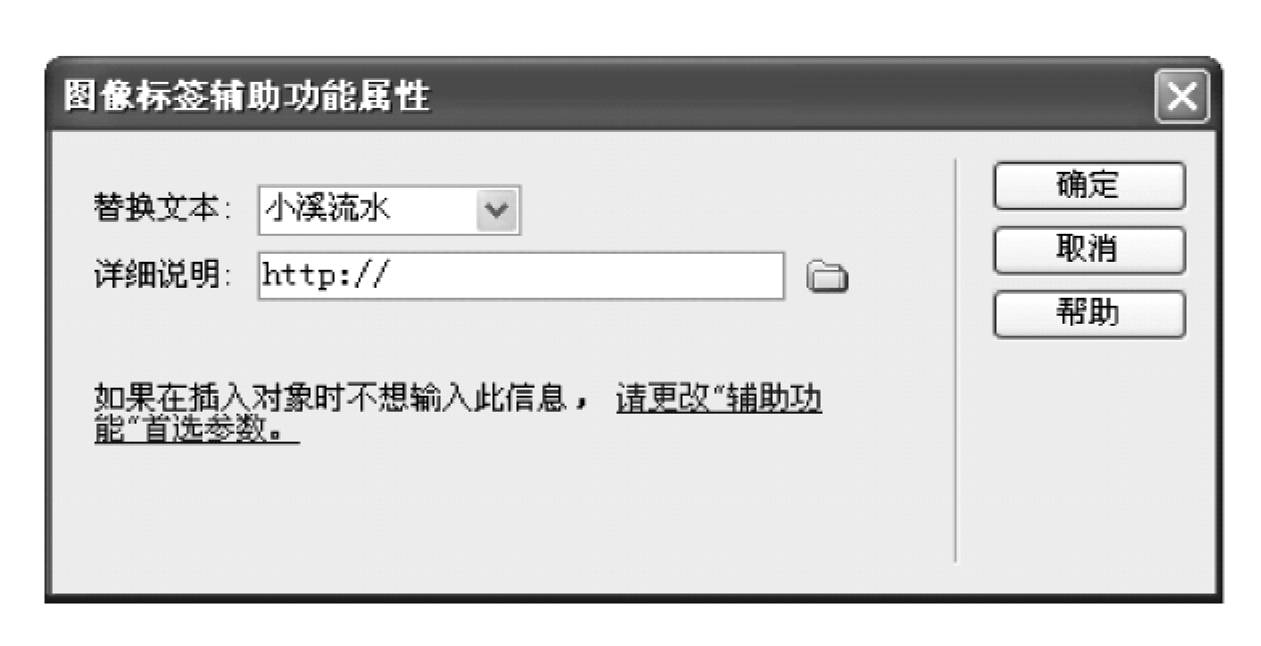


图 6.30　“图像标签辅助功能属性”对话框

（5）单击“确定”按钮，图像插入到网页中，然后拖动图像周围的结点对图像大小作适当的调整，使其更适合网页显示的大小。

2．设置图像属性

在页面中插入图像后，有时还需要对图像进行命名、设置图像大小、修改源文件、设置图像说明、对齐方式、边距和添加边框等操作，这时可以选中图像，通过如图 6.31 所示的“图

像”属性面板来设置。

图 6.31 “图像”属性面板

“图像”属性面板中主要参数的含义如下。

(1) “图像”文本框：用于为图像进行命名，以便使用脚本时对其进行控制或通过定义 CSS 样式来改变图像的显示。

(2) “宽”和“高”文本框：用于设置图像的大小，默认度量单位为像素。当编辑窗口中的图像大小与原始图像大小不一致时，将在文本框右侧显示图标，单击该图标将恢复图像原始大小。

(3) “源文件”文本框：用于显示文件的位置，如果要用新图像替换原始图像，在“源文件”文本框中重新输入要插入图像的位置或单击其后的按钮，在打开的“选择图像源文件”对话框中重新选择其他图像即可。

(4) “替换”下拉列表框：用于设置图像的简短描述文本。在浏览该网页时，当鼠标光标移动到图像上或不能正常显示图像时会显示该文本。

(5) “垂直边距”和“水平边距”文本框：用于设置图像与文本在垂直方向及水平方向的距离。“垂直边距”用于设置图像顶部和底部的边距；“水平边距”用于设置图像左侧和右侧的边距。

(6) “边框”文本框：用于设置边框的宽度，其单位为像素，通常设置为 0。

(7) “对齐”下拉列表框：用于设置同一行上的图像与文本的对齐方式。默认值是基线对齐，是指将文本基准线对齐图像底端。

(8) 亮度和对比度按钮：用于调整图片的明暗度，单击按钮将打开“亮度/对比度”对话框，在“亮度”栏中拖动滑块可调整图像的明暗度；在“对比度”栏中拖动滑块可调整图像的对比度。若选中“预览”复选框，则在调节明暗度和对比度时可看到页面中图像的变化。

3. 创建鼠标经过图像

鼠标经过图像由原始图像和鼠标经过图像组成。在浏览器中查看网页中，当鼠标光标经过图像时图像变为鼠标经过图像，移开鼠标光标后图像又还原到原始图像。

例 6.2 创建图片经过对象，过程如下：

(1) 新建空白文档命名为 Example6-2.html，将插入点定位到要创建鼠标经过图像的位置，选择“插入”→“图像对象”→“鼠标经过图像”命令，打开“插入鼠标经过图像”对话框，如图 6.32 所示。

(2) 在“图像名称”文本框中输入图像名称，单击“原始图像”文本框后的“浏览”按钮。

(3) 使用同样的方法设置鼠标经过图像。

(4) 选中“预载鼠标经过图像”复选框使网页加载完成后将鼠标经过图像预载到本地电脑的缓存中，以避免图像显示延迟。



图 6.32　“插入鼠标经过图像”对话框

(5) 在“替换文本”文本框中输入对图像的描述。在“按下时，前往的 URL”文本框输入当单击鼠标经过图像时跳转到的网页，可以空着，设置完后如图 6.33 所示。



图 6.33　完成鼠标经过图像的设置

(6) 单击“确定”按钮，完成鼠标经过图像的创建，保存网页并预览，鼠标不经过的图如图 6.34 所示，鼠标经过如图 6.35 所示。



图 6.34　鼠标没移动到图片上



图 6.35　鼠标移动到图片上

6.3 超链接使用

在浏览网页的过程中，当单击某些文本或图像时即可跳转到其他网页中，这些文本或图像就是添加了超链接的文本或图像。超链接能够实现页面与页面之间的跳转，从而将网站中的每个页面链接起来。

超链接是由源端点和目标端点两部分组成。其中有超链接的一端称为源端点（响应鼠标单击操作的图像或文本），跳转到的页面称为目标端点（打开的新页面）。在网页中可以添加文本超链接、图像热点超链接、电子邮件超链接、锚超链接和空超链接。

6.3.1 添加基本超链接

基本超链接包括文本超链接和图像超链接，这两种超链接是网页中最常见的超链接方式。选择文本或图像后，在“属性”面板中进行相应设置即可完成链接的添加。

文本超链接的“属性”面板中的“目标”下拉列表框中各选项的含义如下。

(1) _blank 选项：单击超链接文本后，目标端点网页会在一个新窗口中打开。

(2) _parent 选项：单击超链接文本后，在上一级浏览器窗口中显示目标端点网页，该方式在框架网页中比较常用。

(3) _self 选项：单击超链接文本后，在当前浏览器窗口中显示目标端点网页，即替换掉原来的网页。这是 Dreamweaver 的默认设置，如果需要的打开方式就是该方式，可以不在“目标”下拉列表框中进行选择。

(4) _top 选项：单击超链接文本后，在最顶层的浏览器窗口中显示目标端点网页。

下面以添加文本超链接为例进行介绍，其操作步骤如下：

(1) 在网页中选择要创建超链接的文本“This is a hyperlink!”。单击“属性”面板中的“链接”如图 6.36 所示。

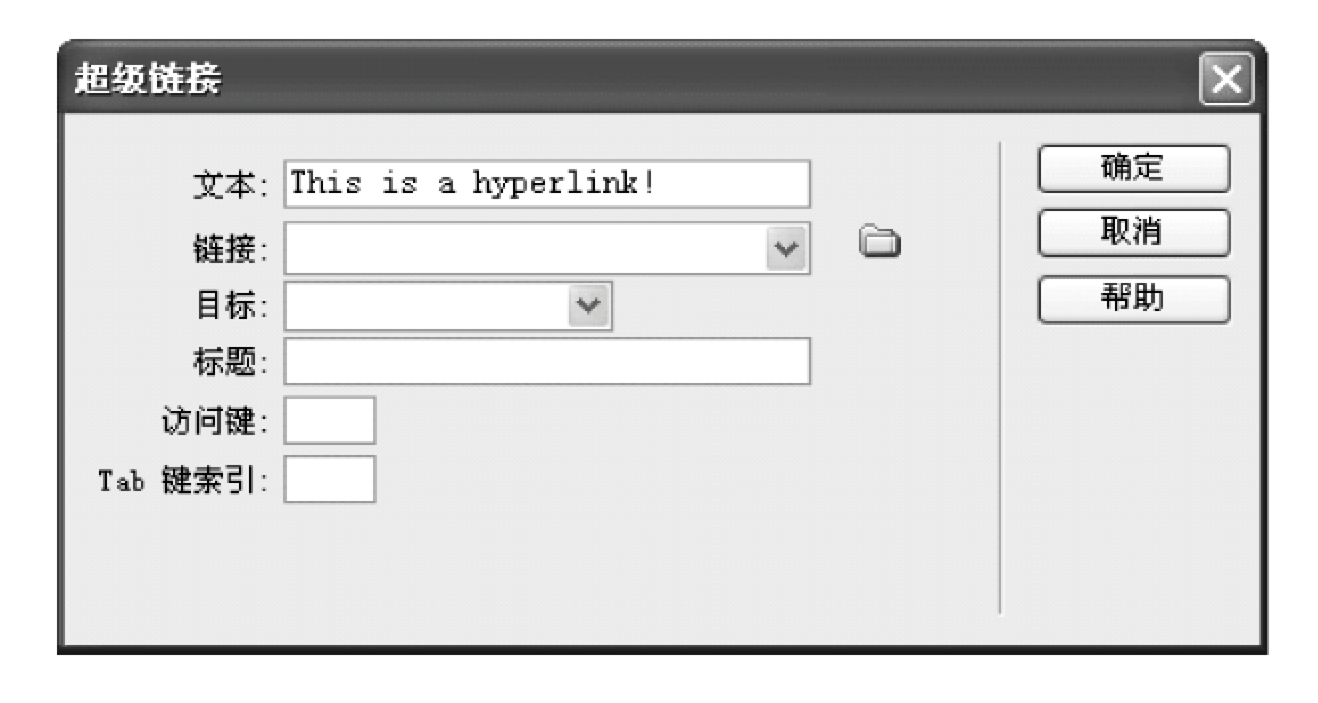


图 6.36 “超级链接”对话框

(2) 可填充图 6.36 各项内容，也可直接单击“确定”按钮，单击“确定”按钮后可以通过属性面板进行设置，如图 6.37 所示。

(3) 单击“确定”按钮关闭“选择文件”对话框。

(4) 在“属性”面板中的“目标”下拉列表框中选择链接网页打开的方式，即可完成文本

链接的创建，如图 6.37 所示。



图 6.37　选择打开目标端点网页的方式(图换掉)

6.3.2　添加图像热点超链接

图像热点超链接是指在一幅图像上的不同区域设置多个超链接。在添加图像热点超链接时，首先需选择图像，使用如图 6.38 所示的“属性”面板左下角的热点创建工具进行热点区域的创建，然后分别选择所创建的热点区域，在“链接”文本框中输入目标端点网页的路径及名称即可。

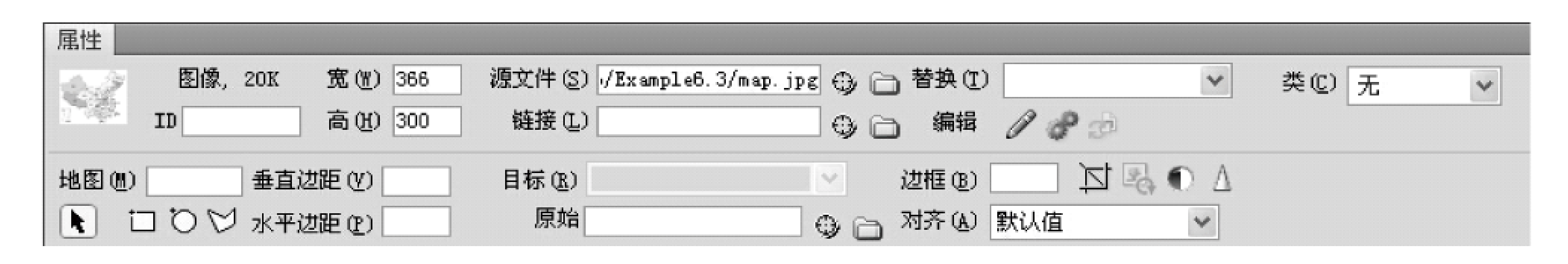


图 6.38　“图像”属性面板

例 6.3　为地图上河南的区域创建热点，链接到 henan.html。

(1) 选择图像，在“属性”面板中选择一个热点创建工具，这里选择“多边形热点工具”。

(2) 按住鼠标左键不放并拖动鼠标光标在图像中的树叶轮廓上绘制出需添加的超链接的多边形热点区域。

(3) 在“属性”面板的“链接”文本框中输入要链接网页的 URL 地址，在“目标”下拉列表框中选择打开目标页的方式，如图 6.39 所示。



图 6.39　为热点区域设置属性

(4) 用同样的方法可以设置其他热点区域，然后保存网页。

6.3.3　添加电子邮件的超链接

使用电子邮件超链接可以方便地进行电子邮件的发送，便于网站管理者收集用户对网站的反馈信息。添加电子邮件超链接的方法很简单，只需将鼠标光标定位到要显示电子邮件超链接文本的位置，在插入栏“常用”Tab 页中单击“电子邮件链接”按钮或选择“插入”→

“电子邮件链接”命令，在打开的“电子邮件链接”对话框的“文本”文本框中输入要显示的电子邮件链接文本，在 E-mail 文本中输入要连接的邮箱地址，单击“确定”按钮即可。

6.3.4 添加锚链接

锚链接的功能是实现源文件文档与目标文档中指定部分的快速跳转。它可以跳转到其他网页中的指定位置，也可以跳转到当前网页中的指定位置。锚链接的添加分为创建命名锚记和创建链接两部分，命名锚记可以理解为目标端点，不过它不是整个网页，而是网页的某一个位置；创建链接命名锚记即创建对应的链接端点。添加锚链接的操作步骤如下：

(1) 将插入点定位到要创建命名锚记的位置或选择要指定的命名锚记的文本，选择“插入”→“命名锚记”的命令，打开“命名锚记”对话框，在“锚记名称”文本框中输入锚链接的名称，如图 6.40 所示。

(2) 单击“确定”按钮完成命名锚记的创建，在定位点即出现锚记符号。

(3) 选择要添加锚链接的文本或图像，在“属性”面板中的“链接”文本框中输入“#”符号，再输入命令锚记的名称。如果目标端点与锚记不在同一个网页中，则应先写上网页的路径及名称，然后加上前缀“#”和锚记名称，在“目标”下拉列表框中输入打开网页的方法，完成链接命名锚记的创建。



图 6.40 “命名锚记”对话框

6.3.5 添加空超链接

空超链接是指未指定目标端点的链接，它用于为页面上的对象附加行为。如果要为网页文本或图像添加脚本代码实现一些特殊的功能，则需要添加空超链接。其方法为：选择要添加空超链接的文本或图像后，在“属性”面板中的“链接”下拉列表框或“链接”文本框中输入“#”符号即可。

6.4 网页布局设计

6.4.1 使用表格布局网页

使用表格进行网页布局是以前最流行的网页布局方式，但由于表格的一些特殊缺点(如加载表格式网页时必须将整个表格的内容加载完成后才能显示内容以致严重影响网页的打开速度等)，使用表格进行网页布局将会越来越少，但使用表格格式化数据的显示仍会是表格的一个重要的应用，下面讲解表格的相关知识。

1. 创建表格

在 Dreamweaver CS5 中可以创建普通表格，也可以创建嵌套表格，下面分别进行讲解。

（1）创建普通表格

在 Dreamweaver CS5 中创建普通表格的操作与在 Word 中创建表格的方法基本相同，其操作步骤如下：

① 将插入点定位到需创建表格的位置。

② 选择“插入”→“表格”命令或在“常用”插入栏中单击“表格”按钮，打开“表格”对话框。

③ 在“行数”“列数”文本框中分别输入“6”和“4”。

④ 在“表格宽度”文本框中输入表格的宽度，在其后的下拉列表框中选择度量单位。

⑤ 在“边框粗细”文本框中输入表格边框的粗细，如“0”。

⑥ 在“单元格边距”文本框中输入单元格中的内容与单元格边框之间的距离值。这是为了查看其效果，为其设置一个较大的值“6”。

⑦ 在“单元格边距”文本框中输入单元格与单元格之间的距离值。这是为了查看其效果，为其设置一个较大的值“6”，如图 6.41 所示。

⑧ 设置完成后单击“确定”按钮关闭对话框，即完成表格的创建，创建的表格效果如图 6.42 所示。



图 6.41　“表格”对话框

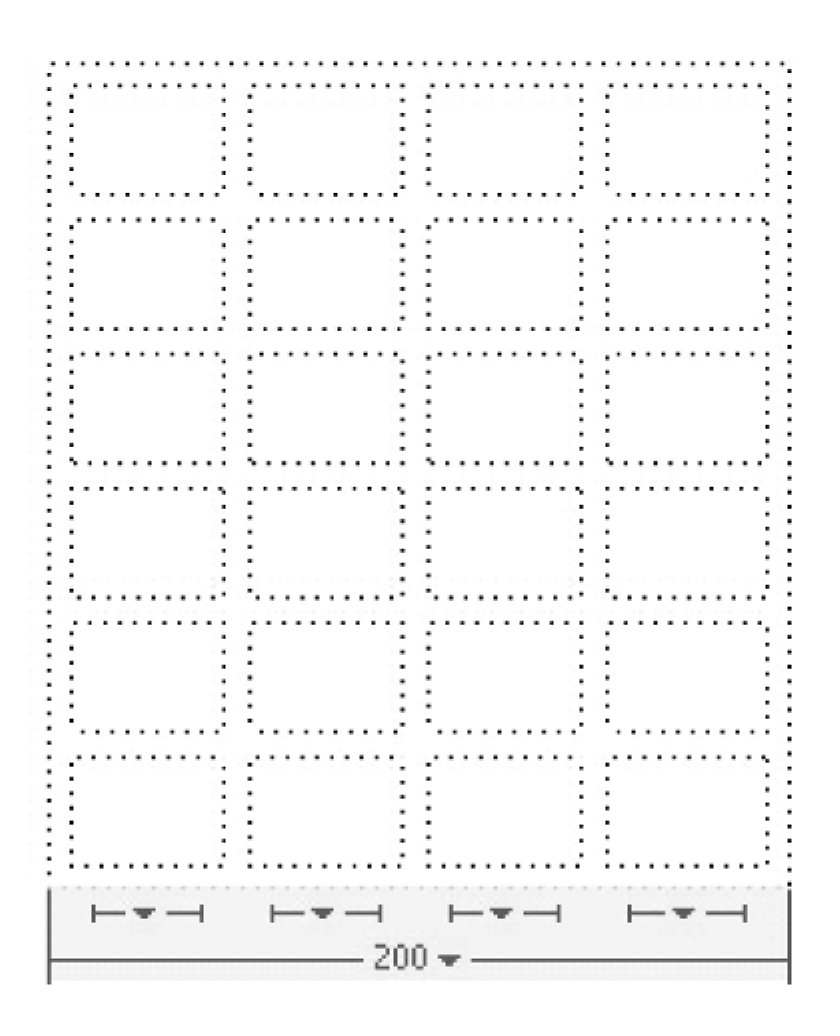


图 6.42　创建的表格

（2）创建嵌套表格

为了使布局更符合实际需要，可以在表格的某些单元格中再插入一个表格，新插入的表格即为嵌套表格。在创建嵌套表格时，需要注意嵌套层数不要太多，否则会影响网页的加载速度。创建嵌套表格的操作步骤如下：

① 将插入点定位到需创建嵌套表格的单元格中。

② 选择“插入”→“表格”命令或在“常用”插入栏中单击 ▦ 按钮，打开“表格”对话框。

③ 在对话框中根据实际情况进行行数、列数、宽度和边框等属性的设置。

④ 设置完成后，如图 6.43 所示，单击“确定”按钮完成嵌套表格的创建，如图 6.44 所示。



图 6.43 “表格”对话框

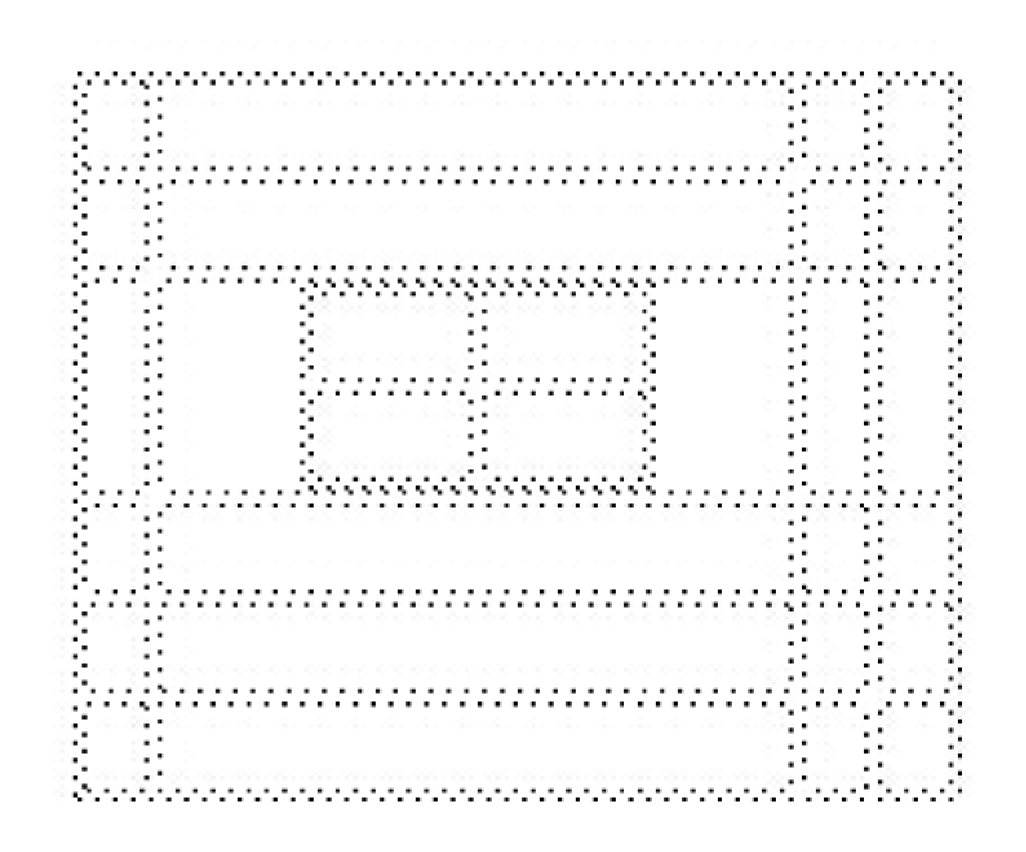


图 6.44 创建的嵌套的表格

2. 编辑表格

除了使用嵌套表格可以让表格布局更加合理外，有时也需要对表格中的单元格进行拆分、合并或删除等操作，在执行这些操作之前，应先选择要操作的对象，如整个表格、某行或某几个单元格等。

3. 设置表格和单元格属性

插入表格后，可在“属性”面板中对表格或单元格的属性进行设置。

(1) 设置表格属性

选中表格的“属性”面板，如图 6.45 所示，可对表格的属性进行设置，其中各项的参数的含义如下：

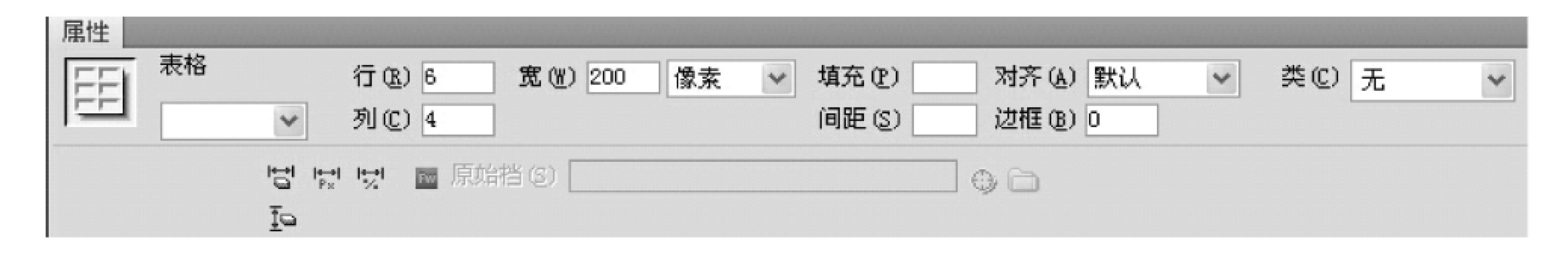


图 6.45 “属性”面板

- “表格”下拉列表框：为表框进行命名，可用于脚本的应用或定义 CSS 样式。
- “行”和“列”文本框：设置表格的行数或列数。
- “宽”文本框：设置表格的宽度，在其后的下拉表框中选择度量单位，如“像素”或“百分比”。
- “填充”文本框：设置单元格边界和单元格内容之间的距离，在“表格”对话框中对应“单元格边距”文本框。

- “间距”文本框：设置相邻单元格之间的距离，在“表格”对话框中对应“单元格间距”文本框。
- “对齐”下拉列表框：设置表格与文本或图像等网页元素之间的对齐方式，只限于和表格同段落的元素。
- “边框”文本框：设置边框的粗细，通常设置为 0 ，如果需要边框，通常通过定义 CSS 样式来实现。
- “类”：设置表格应用的 CSS 格式。

(2) 设置单元格属性

除了设置表格的属性外，通常还需要对表格中的单元格属性进行设置，如单元格的背景颜色、背景图像、单元格中的文本对齐方式等。

选中表格中的单元格(可以同时选择多个单元格，连续或不连续的皆可)后，在“属性”面板中即可对其属性进行设置。

单元格“属性”面板(如图 6.46 所示)上半部分与选中文本时的“属性”面板相同，主要用于设置单元格中的文本的属性，下半部分主要用于设置单元格的属性，其中各项参数的含义如下。



图 6.46　“属性”面板

- 按钮：单击该按钮可合并选中的单元格。
- 按钮：单击该按钮课进行单元格的拆分操作。
- “水平”下拉列表框：用于设置单元格中内容在水平方向上的对齐方式，包括“左对齐”、“居中对齐”、“右对齐”和“默认”4 个选项。
- “垂直”下拉列表框：用于设置单元格中内容在垂直方向上的对齐方式，包括“顶端”、“居中”、“底部”、“基线”和“默认”5 个选项。
- “宽”文本框：设置单元格的宽度，如果直接输入数字，则默认度量单位为“像素”，如果要以“百分比”作为度量单位，则应在输入数字的同时输入“%”符号，如“90%”。
- “高”文本框：设置单元格的高度，通常无须进行设置。
- “不换行”复选框：选中该复选框可以防止换行，使给定的单元格中的所有文本都在一行上。
- “标题”复选框：可以将所选的单元格格式设置为表格标题单元格(也可通过“表格”对话框中的“页眉”栏进行设置)。默认情况下，表格标题单元格的内容为粗体并且居中。
- “背景颜色”文本框：用于设置单元格的背景颜色。

6.4.2 使用 AP 元素布局网页

AP 元素在早期的 Dreamweaver 软件中称为层，在 Dreamweaver CS5 中对其功能进行了增强，为此 Adobe 公司为其取了一个更贴切的名称——AP 元素。

AP 元素最大的特点是具有极高的灵活性，用户可以将 AP 元素放置在页面中的任何位置，且 AP 元素可以重叠，也可以隐藏或显示，在制作一些特殊效果时是非常有用的。

1. 创建 AP 元素

AP 元素的创建很简单，像在 Photoshop、Fireworks 等图形软件中绘制矩形一样，可以理解为 AP 元素是一个灵活移动的矩形框。

(1) 创建普通的 AP 元素

创建普通的 AP 元素的操作步骤如下：

① 将“插入”栏切换到“布局”插入栏，单击“绘制 AP 元素”按钮，如图 6.47 所示。

② 在编辑窗口中任意位置按下鼠标左键不放进行拖动即可创建 AP 元素。



图 6.47 单击“绘制 AP 元素”按钮

(2) 创建嵌套 AP 元素

AP 元素与表格一样，也可以进行嵌套，而且可以嵌套多层。嵌套的 AP 元素之间有非常紧密的关系，如拖动外层 AP 元素时可以同时移动内层嵌套 AP 元素，而且可以同时设置其隐藏属性等。

创建嵌套 AP 元素的方法很简单，其操作步骤如下：

① 将插入点定位到需要创建嵌套的 AP 元素中。

② 选择“插入”→“布局对象”→AP Div 命令即可在现有 AP 元素中创建一个 AP 元素。

③ 将插入点定位到新创建的嵌套 AP 元素中。

④ 在“布局”插入栏中单击“绘制 AP 元素”按钮，在嵌套 AP 元素中进行拖动，可以绘制出一个新的嵌套 AP 元素。

2. 编辑 AP 元素

要对 AP 元素进行编辑，仍需要先选择 AP 元素，然后再进行 AP 元素的大小调整，对齐 AP 元素，AP 元素的堆叠顺序设置及 AP 元素的显示与隐藏设置等操作。

选择“窗口”→“AP 元素”命令打开“AP 元素”面板，如图 6.48 所示。在“AP 元素”面板中显示了网页中所有的 AP 元素及各个 AP 元素之间的关系，在“AP 元素”面板中可以选择 AP 元素，设置 AP 元素的显示属性，设置 AP 元素的堆叠顺序及重命名 AP 元素。另外，在编辑窗口及“属性”面板中也可对 AP 元素进行一些编辑，如设置隐藏或显示属性等，下面分别讲解其设置方法。

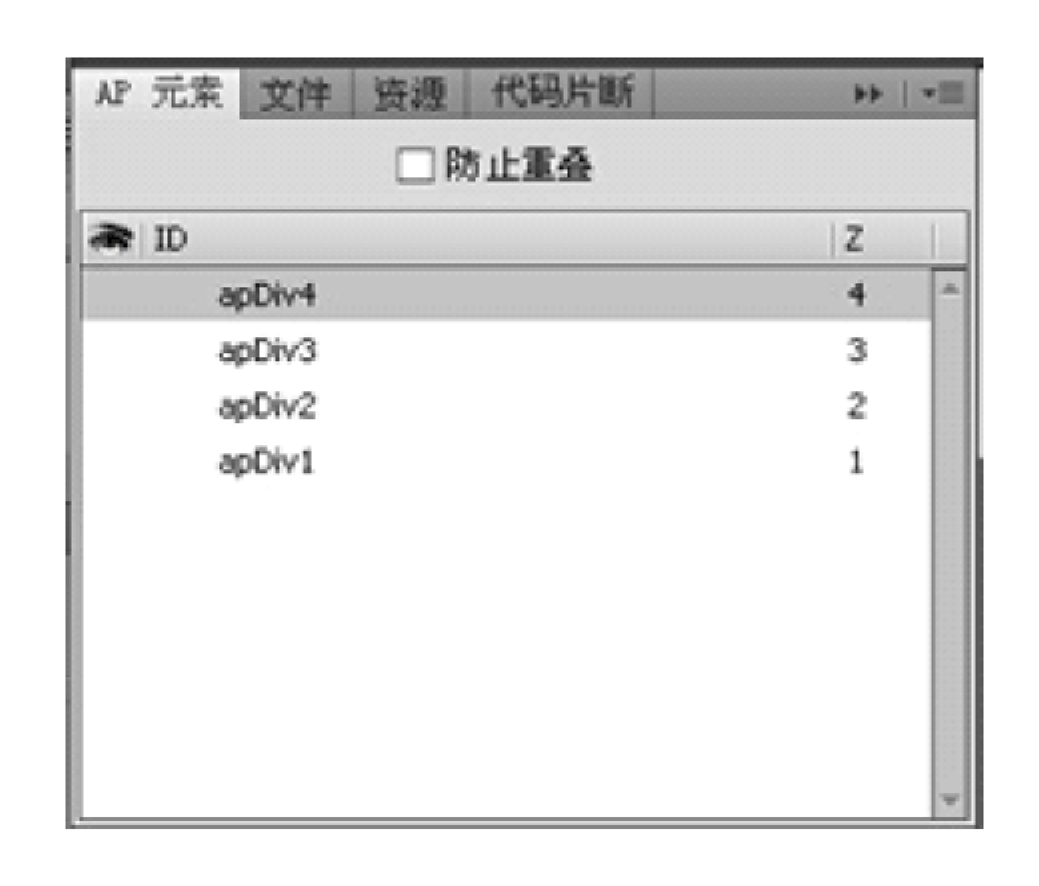


图 6.48　“AP 元素”面板

（1）选择 AP 元素

在“AP 元素”面板及编辑窗口中选择 AP 元素的操作步骤如下：

① 打开“AP 元素”面板。

② 单击要选择的 AP 元素的名称，选择该 AP 元素。

③ 按住 Shift 键后依次在“AP 元素”面板中单击需选中 AP 元素的名称即可选中多个 AP 元素，如图 6.49 所示。



图 6.49　选择单个 AP 元素

④ 在编辑窗口中单击要选择的 AP 元素的边框即可选择单个 AP 元素。

⑤ 按住 Shift 键后依次在需要选中的 AP 元素中或 AP 元素边框上单击可选中多个 AP 元素。

（2）调整 AP 元素的大小

选中要调整大小的 AP 元素后，在编辑窗口中即可对其进行大小的调整，如果要是选中的 AP 元素具有相同的高度或宽度，则可以选择相应的菜单命令来完成，其操作步骤如下：

① 将鼠标指针移至要调整大小的 AP 元素的边框上，边调整大小，边观察“属性”面板中动态显示的“宽”和“高”数值并进行拖动，至需要大小后释放鼠标即可。

② 选中左侧的 AP 元素，在“属性”面板的“宽”、“高”文本框中分别输入相应数值。

③ 将鼠标指针移动到右侧的 AP 元素中单击，再将鼠标指针移动到左侧的 AP 元素中，按住 Shift 键并单击鼠标左键，完成两个 AP 元素的选择。

④ 选择“修改”→“排列顺序”→“设成宽度相同”命令。

（3）对齐 AP 元素

对齐 AP 元素是网页制作常遇到的问题，选择 AP 元素后，选择相应的菜单命令即可进行 AP 元素的对齐操作，其操作步骤如下：

① 选择需对齐的所有 AP 元素(先选择右侧的 AP 元素，再选择左侧的 AP 元素)。

② 选择“修改”→“排列顺序”→“左对齐”命令，完成 AP 元素的左对齐操作。

③ 选择“修改”→“排列顺序”→“对齐下缘”命令，完成 AP 元素下缘对齐操作。

（4）移动 AP 元素

选择需移动的 AP 元素后，将鼠标指针移动到 AP 元素边框上，当鼠标指针变为 形状时，按住鼠标左键不放进行拖动，到需要的位置后释放鼠标即可。

（5）设置 AP 元素的堆叠顺序

AP 元素可以重叠，就如 Photoshop 或 Firework 等图形图像软件中的“图层”一样，不同的重叠顺序可能导致不同的效果，因此，设置合理的堆叠顺序是非常有必要的。

在“AP 元素”面板或属性面板中可以进行 AP 元素堆叠顺序的设置，其他操作步骤如下：

① 打开“AP 元素”面板，选择所需的 AP 元素，按住鼠标左键不放将其拖动到所需的位置后释放鼠标即可。

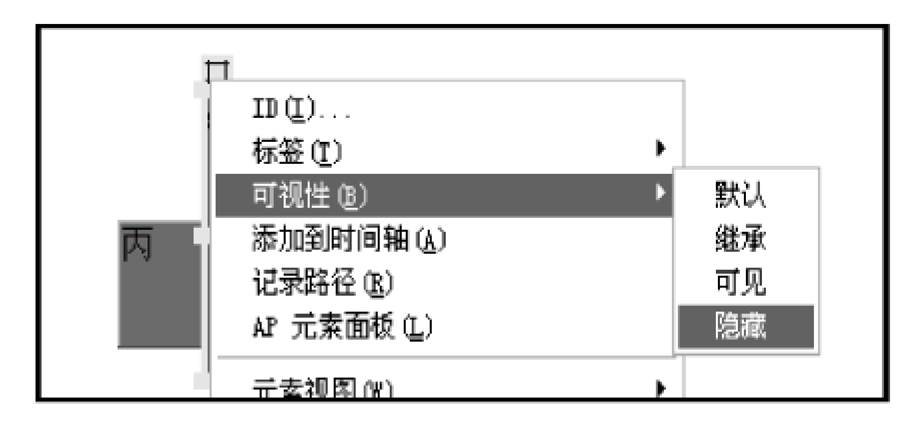
② 在编辑窗口或“AP 元素”面板中选择需改变堆叠顺序的 AP 元素后，在“属性”面板中的“Z 轴”文本框中输入所需的数值，大于原数字可将该 AP 元素在堆叠顺序中上移，小于原数字可将该 AP 元素在堆叠顺序中下移，完成输入后按 Enter 键确认。

（6）改变 AP 元素的可见性

设置 AP 元素的可见性就是设置 AP 元素的显示与隐藏，可以动手设置 AP 元素的可见性，也可以通过编写 JavaScript 代码来实现 AP 元素可见性的设置。

设置 AP 元素的可见性的具体操作步骤如下：

① 选择要设置可见性的 AP 元素后，在选择的 AP 元素上单击鼠标右键，在弹出的快捷菜单中选择“可视性”|“隐藏”命令，再在编辑窗口的空白区域单击鼠标左键即可隐藏 AP 元素，如图 6.50 所示。



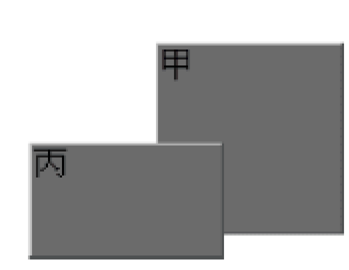


图 6.50 隐藏 AP 元素

② 在“AP 元素”面板中选择隐藏的 AP 元素，再单击前面的 图标，使其变为 图标，即可显示 AP 元素，如图 6.51 所示。

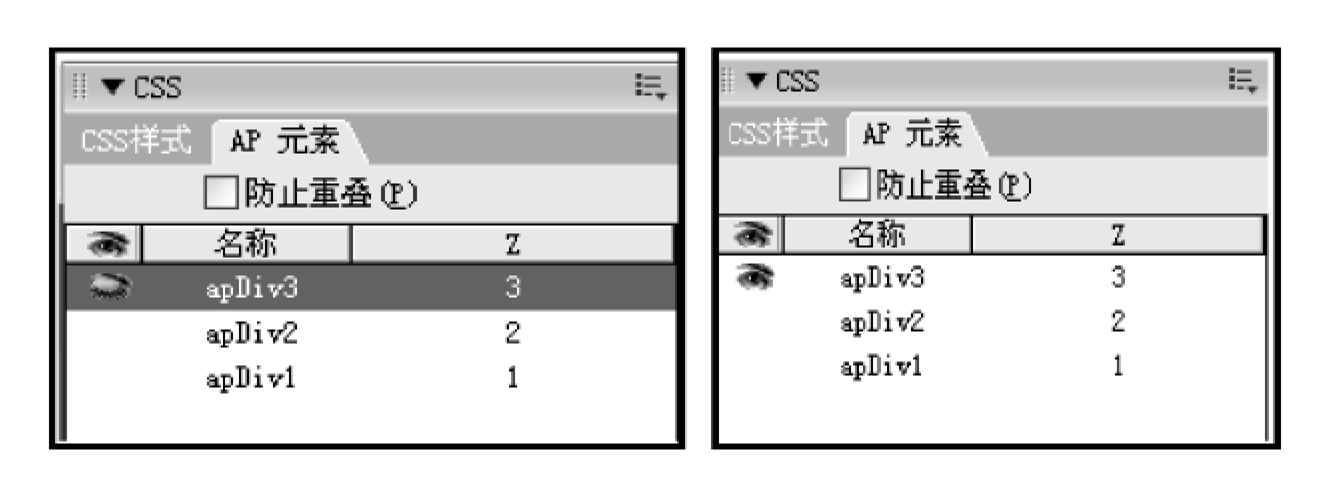


图 6.51　显示 AP 元素

3. AP 元素的属性设置

创建 AP 元素后，可以在“属性”面板中对 AP 元素的属性进行设置，下面分别讲解单个 AP 元素和多个 AP 元素的属性设置方法。

（1）设置单个 AP 元素的属性

选择要设置属性的单个 AP 元素，其“属性”面板如图 6.52 所示。面板中各设置项含义如下。



图 6.52　单个 AP 元素的“属性”面板

- CSS-P 元素下拉列表框：可为当前 AP 元素命名，该名称可在脚本中引用，如通过编写脚本实现 AP 元素的显示或隐藏等。
- “左”文本框：设置 AP 元素左边相对于页面左边或父 AP 元素左边的距离。
- “上”文本框：设置 AP 元素顶端相对于页面顶端或父 AP 元素顶端的距离。
- “宽”文本框：设置 AP 元素的宽度值。
- “高”文本框：设置 AP 元素的高度值。
- “Z 轴”文本框：设置 AP 元素的 Z 轴顺序，也就是设置嵌套 AP 元素在网页中的堆叠顺序，较高值 AP 元素位于较低值的 AP 元素的上方。
- “可见性”下拉列表框：设置 AP 元素的可见性，其中 default 选项为默认值，其可见性由浏览器决定，大多数浏览器会继承该 AP 元素的父 AP 元素的可见性；inherit 选项表示继承其父 AP 元素的可见性；visible 选项表示显示 AP 元素及其内容，而与其父 AP 元素无关；hidden 选项表示隐藏 AP 元素及其内容，与父 AP 元素无关。
- “背景图像”文本框：用于设置背景图像，单击 按钮，在打开的“选择图像源文件”对话框中可选择所需的背景图像。
- “背景颜色”文本框：设置 AP 元素的背景颜色。
- “类”下拉列表框：选择 AP 元素的样式。
- “溢出”下拉列表框：选择当 AP 元素中的内容超出 AP 元素的范围后显示内容的方式，其中：visible 选项表示当 AP 元素中的内容超出 AP 元素的范围时，AP 元素自动向右或向下扩展，使 AP 元素能够容纳并显示其中的内容；hidden 选项表示当

AP 元素中的内容超出 AP 元素范围时，AP 元素的大小保持不变，也不出现滚动条，超出 AP 元素范围的内容将不显示；scroll 选项表示无论 AP 元素中的内容是否超出 AP 元素范围，AP 元素的右端和下端都会出现滚动条；auto 选项表示当 AP 元素中的内容超出 AP 元素范围时，AP 元素的大小保持不变，但是在 AP 元素的左端或下端会出现滚动条，以便使 AP 元素中超出范围的内容能够通过拖动滚动条来显示。

• “剪辑”栏：在该栏中可设置 AP 元素的可见区域。其中“左”、“右”、“上”和“下”4 个文本框分别用于设置 AP 元素在各个方向上的可见区域与 AP 元素边界的距离，其单位为“像素”。

(2) 设置多个 AP 元素的属性

在 Dreamweaver CS5 中可同时选择多个 AP 元素进行相同的属性设置，选择多个 AP 元素后的“属性”面板如图 6.53 所示。



图 6.53 选择多个 AP 元素的“属性”面板

在多个 AP 元素的“属性”面板中部可以设置 AP 元素中文本的样式，大部分属性与单个 AP 元素的“属性”面板相同，不同的是多个 AP 元素的“属性”面板中多了一个“标签”下拉列表框，其中包括 DIV 和 SPAN 两个选项，二者的功能基本相同，但选择 DIV 选项时，通常会独占一行，即在其后不能再添加其他对象，而选择 SPAN 选项则不同，在 SPAN 标签后是可以继续添加其他对象的。

6.4.3 使用框架布局网页

通常在一个浏览器页面中只能显示一个网页文档的内容，如果需要在同一个浏览器窗口中显示多个网页文档的内容，就需要使用框架或 iframe 浮动框架来创建。下面讲解框架网页的制作与设置方法。

1. 创建框架集

框架集是框架网页的总管，它记录了框架网页中包括的框架个数，各个框架的大小和位置以及各个框架对应的网页文档等。

框架网页可以只包含一个框架集，也可以嵌套多个框架集，其操作步骤如下：

(1) 在插入栏点击 的下拉箭头会出现如图 6.54 所示的下拉菜单。

(2) 在图 6.54 菜单中选择要插入的框架，单击选中的项目弹出“框架标签辅助功能属性”对话框，如图 6.55 所示。

(3) 单击“创建”按钮，打开“框架标签辅助功能属性”对话框，在“框架”下拉列表框中选择某个框架，在“标题”文本中输入该框架的标题，通常保持默认设置，如图 6.55 所示。

(4) 单击“确定”按钮关闭对话框，完成框架网页的创建，其效果如图 6.56 所示。

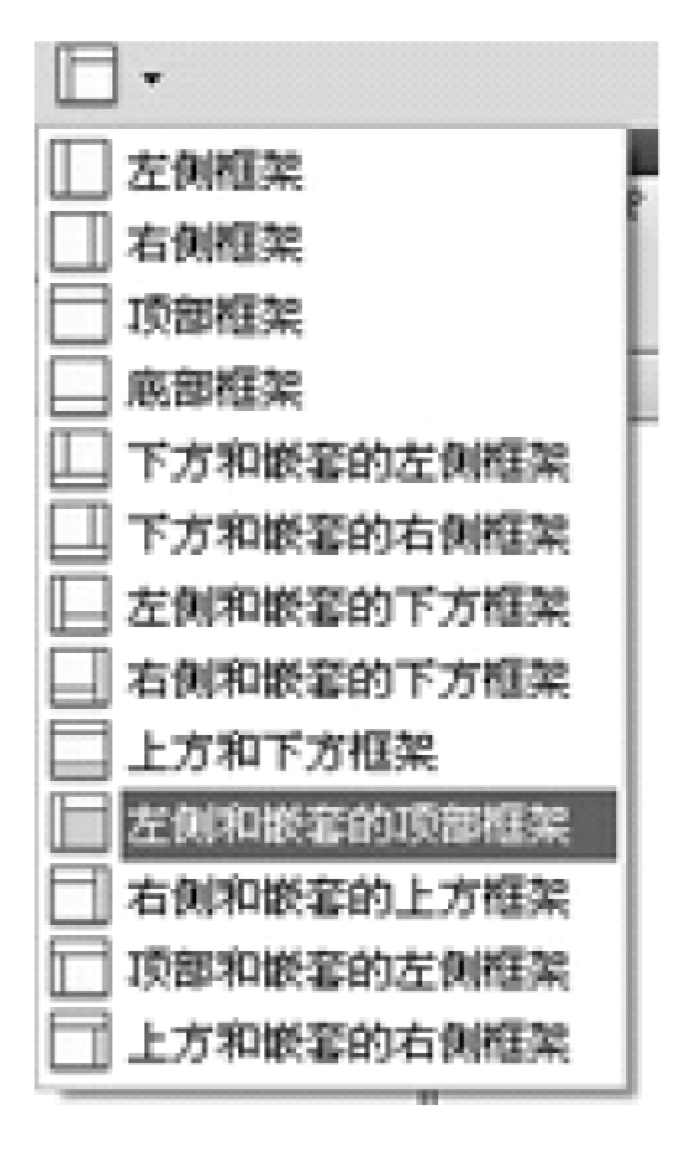


图 6.54　选择框架菜单



图 6.55　"框架标签辅助功能属性"对话框



图 6.56　创建框架

2. 编辑框架

创建框架后，可以选择框架，并进行框架大小的调整等操作。

（1）选择框架或框架集

框架和框架集的选择可在编辑窗口中进行，也可在框架面板中进行。选择框架操作步骤如下：

① 在需选择的框架内单击鼠标左键即可选择该框架，被选择的框架边框以虚线显示。

② 选择"窗口"→"框架"命令显示"框架"面板后，直接在"框架"面板中单击即可选择相应的框架，被选择的框架在面板中以粗黑框显示，如图 6.57 所示。

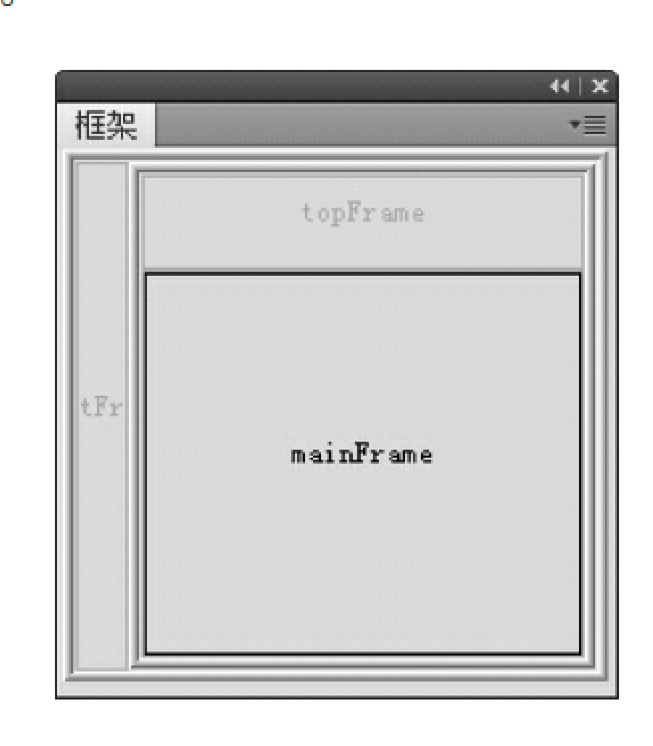


图 6.57　"框架"面板

③ 在编辑窗口中将鼠标指针移动到框架集所在的框架的

边框线上单击,即可选择框架集。

④ 在“框架”画板中将鼠标指针移动到框架集边框上单击即可选择框架集。

(2) 拆分框架

Dreamweaver CS5 默认的框架集可能不适合实际需要,此时可以手动对框架进行拆分,如将某个框架拆分为左右两个框架等,其步骤如下:

① 按住 Alt 键单击要拆分的框架以选中该框架。

② 按住 Alt 键的同时拖动框架至合适的位置后释放鼠标,即完成框架的拆分操作。

(3) 调整框架

在编辑窗口中可以手动拖动框架边框调整框架的大小,但这种调整方法不是很精确,常需要根据在浏览器中的效果进行多次的修正后才能达到满意效果。因此最好的方法是直接在“属性”面板中进行框架大小的设置,其操作步骤如下:

① 选择要设置大小的框架集。

② 在“属性”面板右侧单击要设置大小的框架,再在“列”数值框中输入相应的数值。

③ 使用同样的方法,完成其他框架的大小设置。

(4) 删除框架

若需删除框架,可用鼠标将要删除框架的边框拖至页面外即可。

3. 设置框架的属性

选择框架或框架集后,可在“属性”面板中设置其属性,除了前面讲的设置框架的大小外,还可以设置如名称、源文件、空白边距、滚动特性以及边框特性等。

(1) 设置框架的属性

选择所需设置属性的框架,其“属性”面板如图 6.58 所示。



图 6.58 “框架”属性面板

“属性”面板中各参数含义如下。

- “框架名称”文本框:用于设置框架的名称,可被 JavaScript 程序引用,也可以作为打开链接的目标框架名。
- “源文件”文本框:显示框架网页文档的路径及文件名,单击文本框后的 按钮可在打开的对话框中重新定义。
- “滚动”下拉列表框:设置框架出现滚动条的方式,“是”表示始终显示滚动条;“否”表示始终不显示滚动条;“自动”表示当框架文档内容超出了框架大小时,才会出现框架滚动条;“默认”表示采用大多数浏览器采用的自动方式。
- “不能调整大小”复选框:选中该复选框则不能在浏览器中通过拖动框架边框来改变框架大小。
- “边框”下拉列表框:设置是否显示框架的边框。

• “边框颜色”文本框：设置框架边框的颜色。
• “边框宽度”文本框：输入当前框架中的内容距左右边框间的距离。
• “边框高度”文本框：输入当前框架中的内容距上下边框间的距离。

(2) 设置框架集的属性

选择需要设置属性的框架集，其“属性”面板如图 6.59 所示。

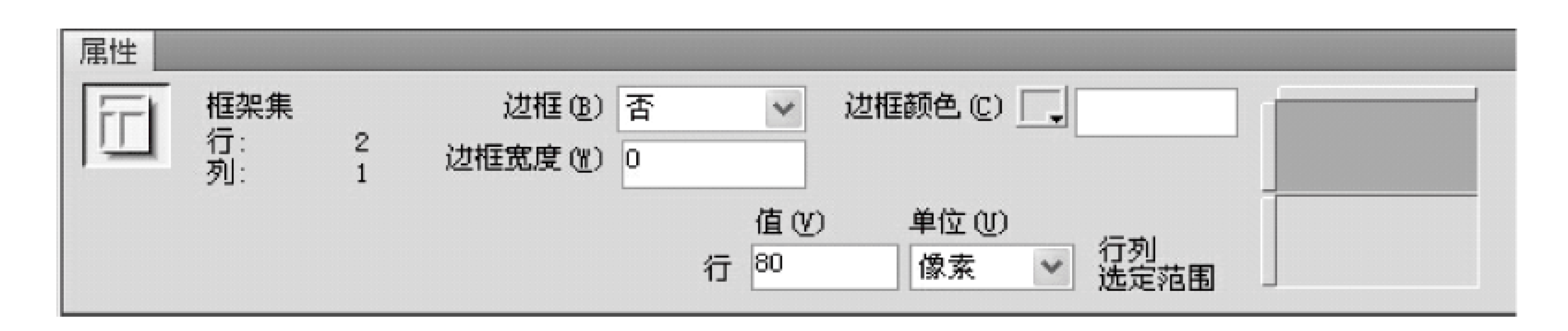


图 6.59　框架集“属性”面板

框架集“属性”面板各设置参数含义和框架“属性”面板基本相同，不同的是在“行”或“列”栏中可设置框架的行或列的宽度(即框架的大小)，在“单位”下拉列表框中选择度量单位后即可输入所需的数值。

4. 保存框架及框架集网页

由于框架集中包括多个网页，因此与一般的网页保存方法略有不同，可以单独保存某一个框架网页文档，也可以单独保存框架集网页文档，另外，也可以同时保存框架集及所有的框架网页的文档，其操作步骤如下：

(1) 将插入点定位到需要保存网页文档的框架中。

(2) 选择“文件”|“保存框架”命令，在打开的“另存为”对话框中的“保存在”下拉列表框中选择保存位置，在“文件名”文本框中输入文件名。

(3) 单击“保存”按钮即可完成框架网页文档的保存。

(4) 选择所需要保存网页文档的框架集。

(5) 选择“文件”|“框架集另存为”命令，在打开的“另存为”对话框的“保存在”下拉列表框中选择保存位置，在“文件名”文本框中输入文件名。

(6) 单击“保存”按钮即可完成框架集网页文档的保存。

(7) 选择“文件”|“保存全部”命令，打开“另存为”对话框。

(8) 在“另存为”下拉列表框中选择保存的路径，在“文件名”文本框中输入网页文档的名称。

(9) 单击“保存”按钮完成框架网页文档的保存，同时打开“另存为”对话框进行其他框架网页文档的保存。

(10) 使用相同的方法完成所有的框架或框架集网页文档的保存。

6.5　使用 CSS 样式表

在 Dreamweaver CS5 中应用 CSS 时，有两种方法，一种是建立 CSS 文件，然后在网页中引用，另一种是直接在网页中建立 CSS 样式，然后应用。

6.5.1 使用 CSS 文件

1. 创建 CSS 文件

在 Dreamweaver CS5 中要创建一个 CSS 文件，在欢迎界面，选择新建 CSS，新建一个 CSS 文件，如图 6.60 所示。

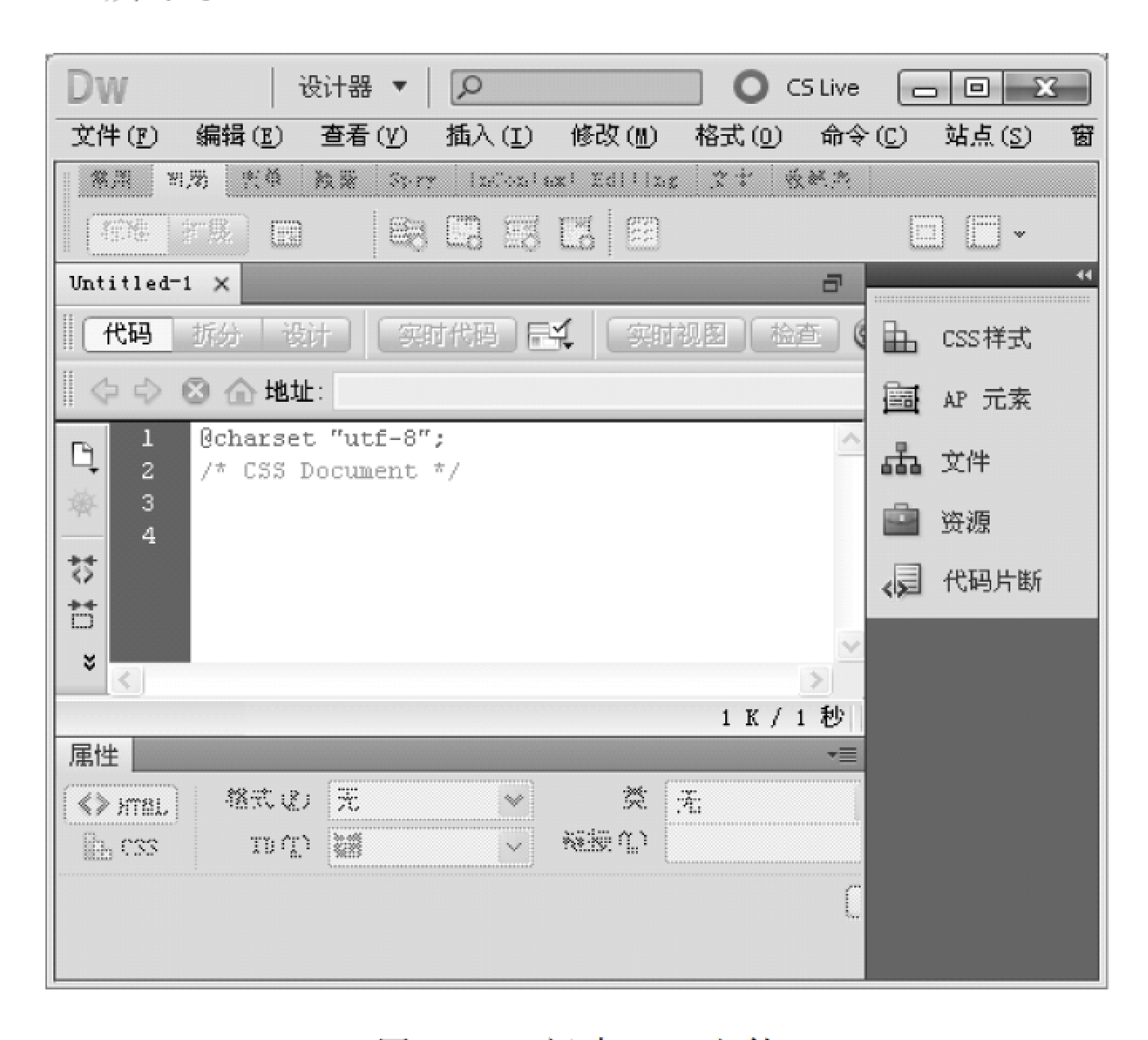


图 6.60　新建 CSS 文件

打开 CSS 样式面板，如图 6.61 所示。在空白区点击鼠标右键，选择“新建”，弹出“新建 CSS 规则”对话框，如图 6.62 所示。



图 6.61　“CSS 样式”面板



图 6.62　“新建 CSS 规则”对话框

单击选择器类型，可以看到选择器类型有4种，如图6.63所示，分别是类（可应用于任何HTML元素），ID（仅用于一个HTML元素），标签（重新定义HTML元素）和复合内容（基于选择的内容），这里选用第一种。输入自选的选择器名称".font"，然后单击确定，弹出"CSS规则定义"对话框，并修改部分属性的值，如图6.64所示，单击"确定"按钮，CSS文件中就会出现如图6.65所示内容。

保存这个文件，并命名为Example6-4.css。



图6.63　输入选择器名称

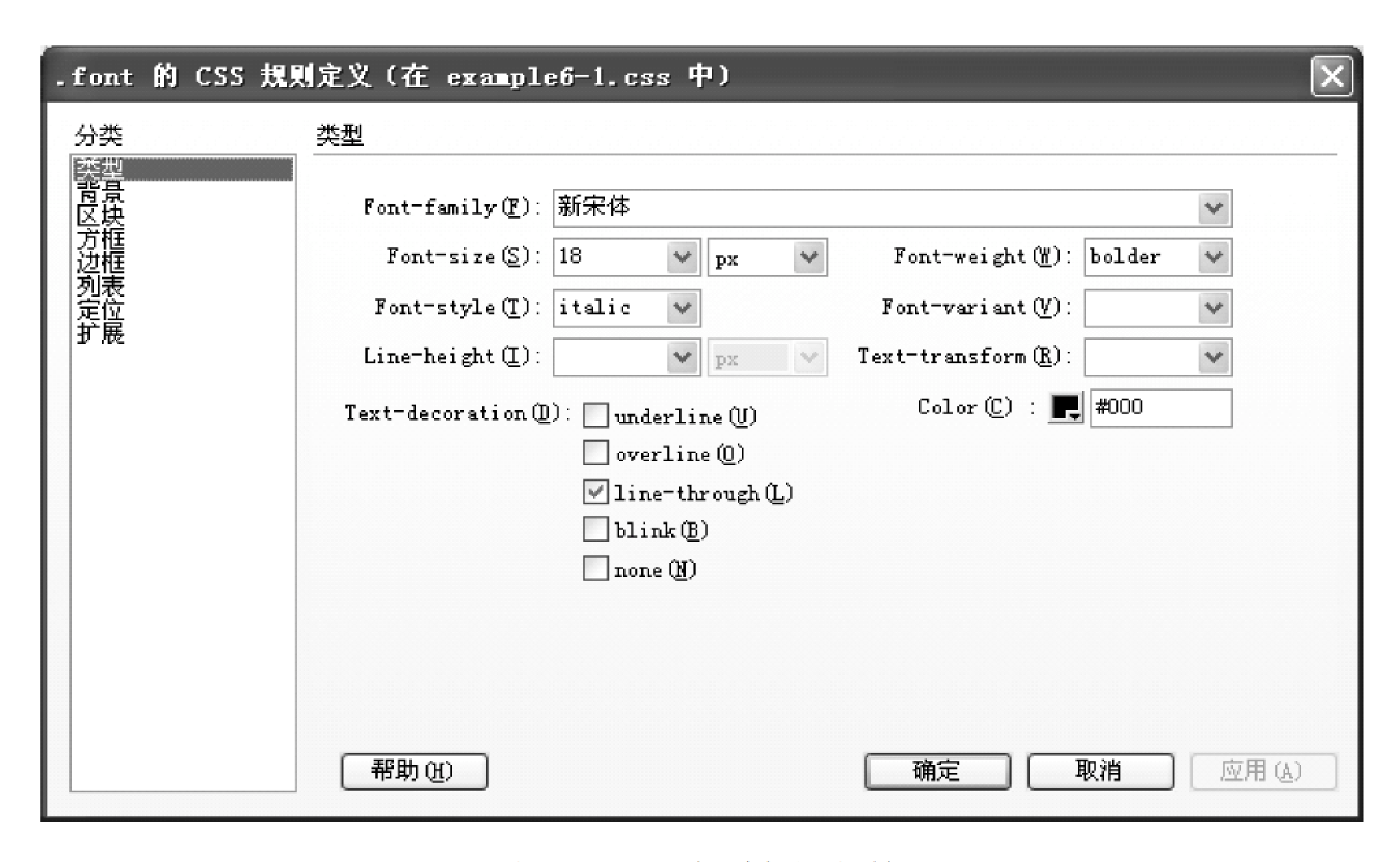


图6.64　CSS规则定义对话框

2. 应用 CSS 文件

例 6.4 应用 CSS 文件的例子。

(1) 建立空白 HTML 文件,并链接已有的 CSS 文件。

新建一个 HTML 空白文档并命名为 Example6-4.html,打开 CSS 面板,选择"附加样式表",如图 6.66 所示,弹出一个"链接外部样式表"对话框,如图 6.67 所示。

```
@charset "utf-8";
/* CSS Document */

.font {
    font-family: "新宋体";
    font-size: 18px;
    color: #000;
    font-weight: bolder;
    text-decoration: line-through;
    font-style: italic;
}
```

图 6.65 CSS 文件内容



图 6.66 选择附加样式表

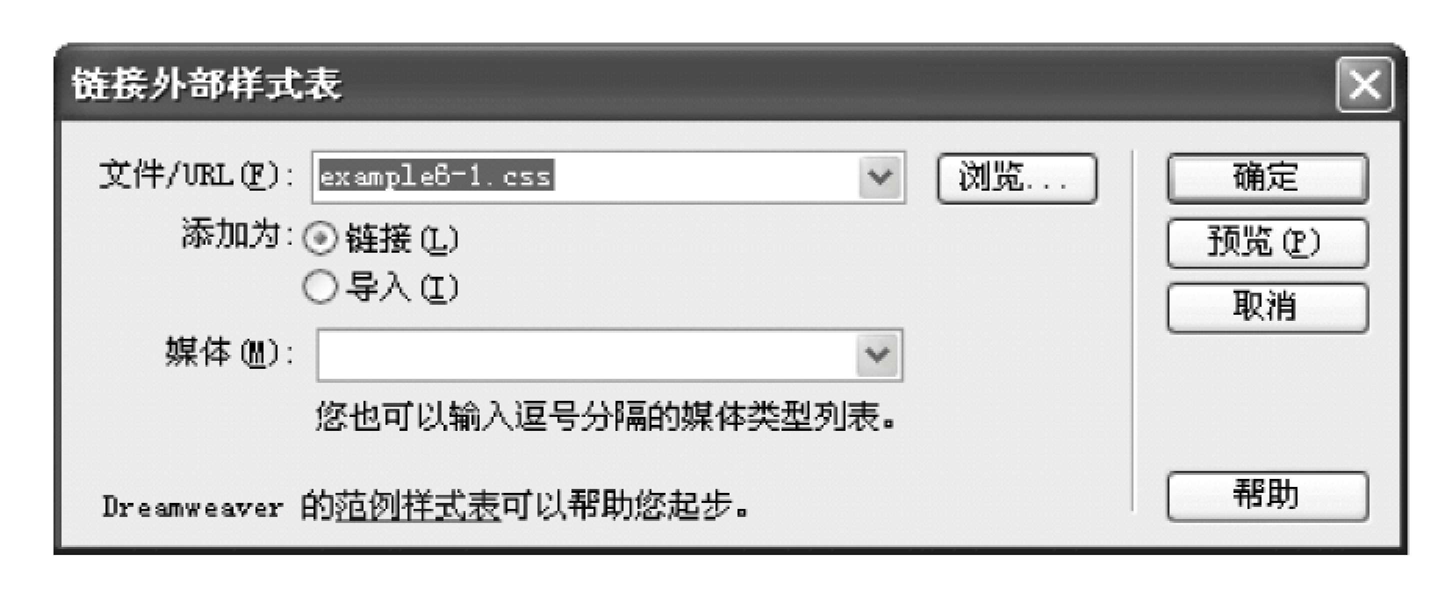


图 6.67 "链接外部样式表"对话框

在图 6.67 所示对话框中选择前面建立的 CSS 文件 example6-4.css,然后单击"确定"按钮,CSS 面板发生了变化,面板中出现了 example6-4.css 中定义的 CSS 样式,如图 6.68 所示。

(2) 在 HTML 文件中放入内容。

在空白网页 example6-4.html 上放置一段文字,在浏览器中的显示如图 6.69 所示。

(3) 对网页中的元素应用 CSS 样式。

在 Dreamweaver CS5 中网页的设计状态下,选中文字,在属性面板中类属性中选择".font",如图 6.70 所示。

选中".font"样式后,网页中文本的格式会发生变化,在浏览器中的显示如图 6.71 所示。



图 6.68　链接了 CSS 文件后的 CSS 面板

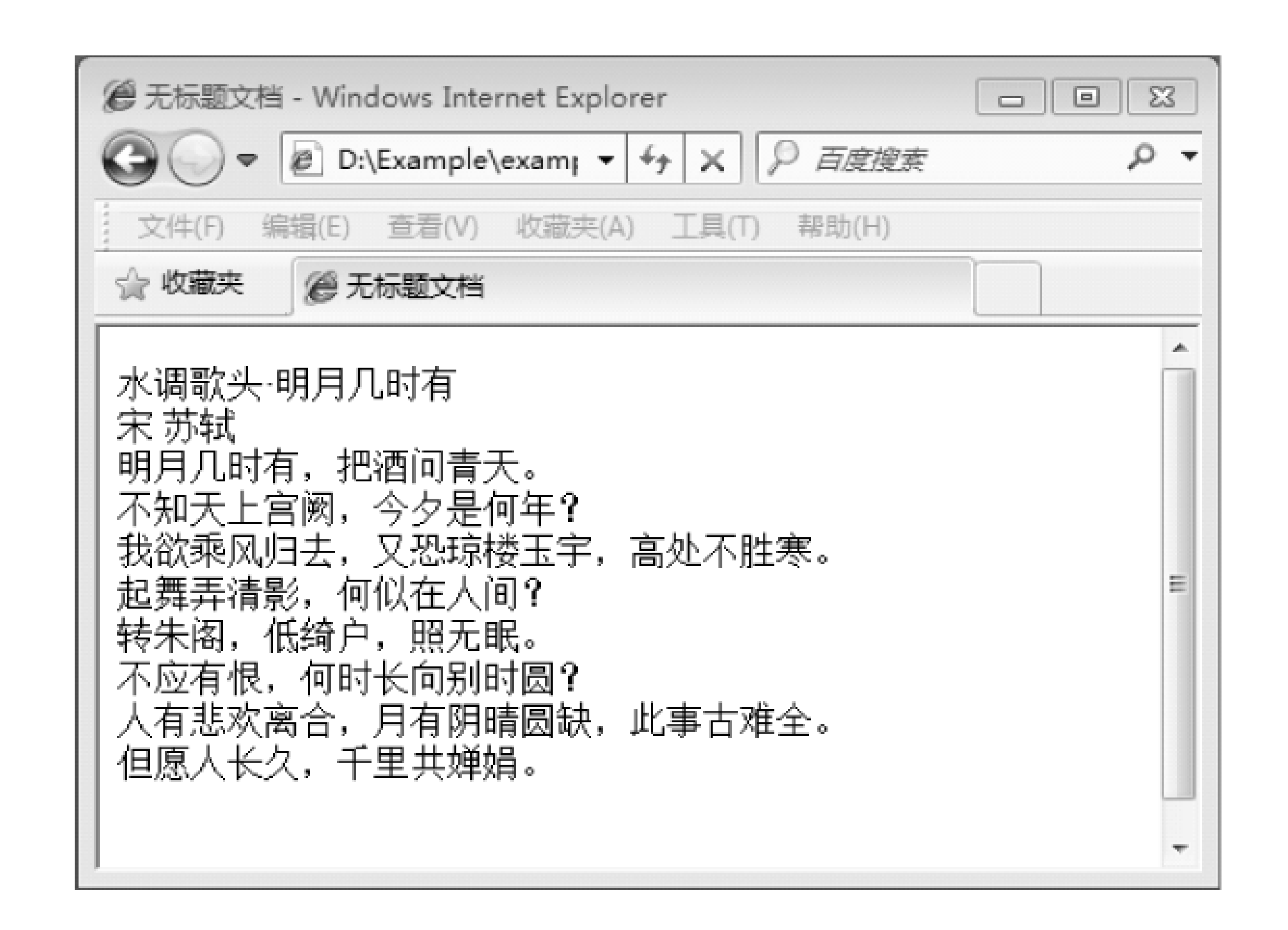


图 6.69　未使用 CSS 样式



图 6.70　未选中的为应用 CSS 样式

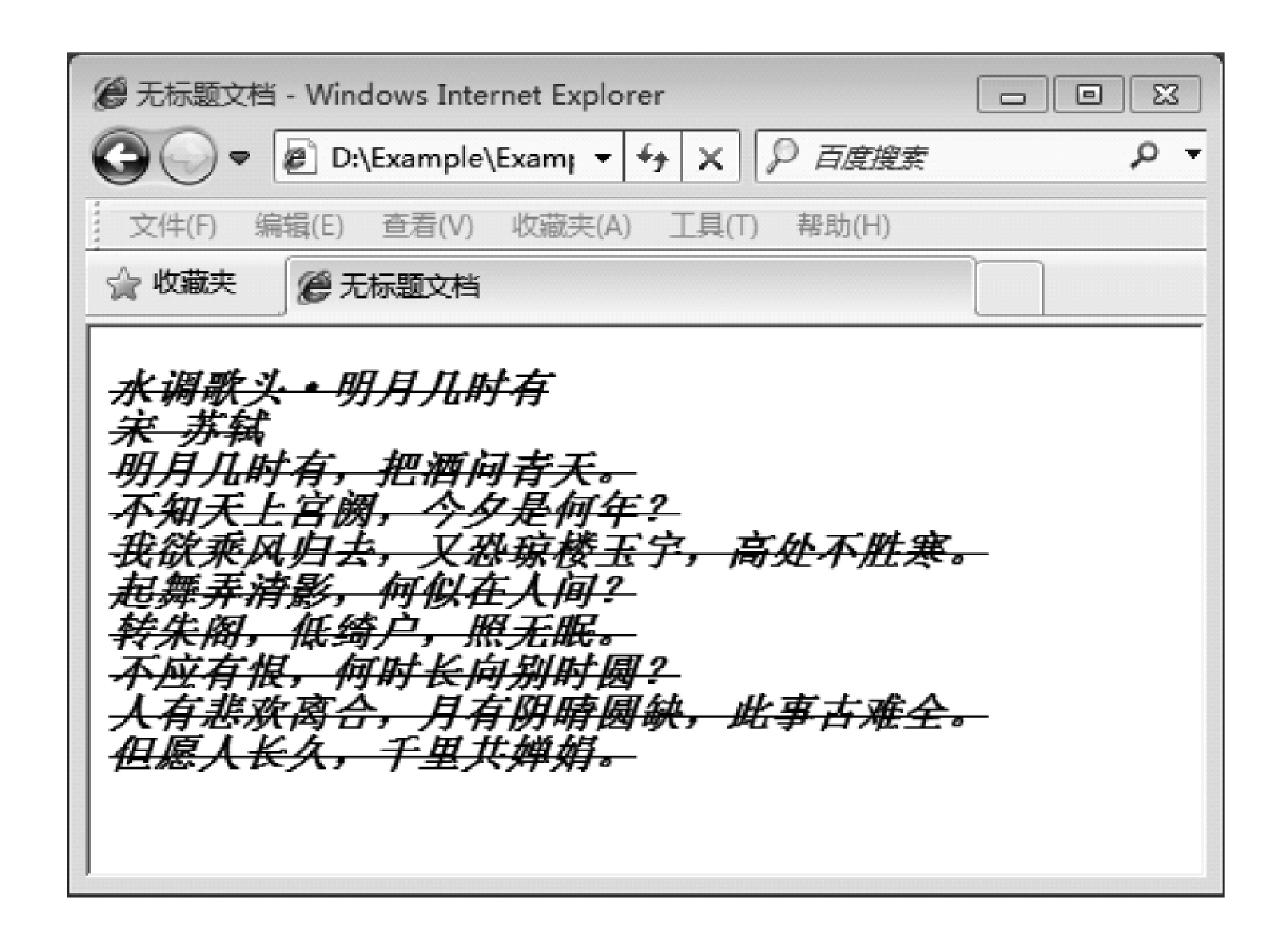


图 6.71　应用过 CSS 样式的效果

6.5.2 网页中的 CSS 样式

例 6.5 在网页中创建并应用 CSS 样式,达到和例 6.4 同样的效果。

新建一个 HTML 文件,打开 CSS 面板,如图 6.72 所示。

在图 6.72 的空白区域单击鼠标右键选择“新建”,过程如 6.5.1 节中一样,结果如图 6.73 所示。

从图 6.73 中可以看出,CSS 样式定义被加入到了网页中,保存网页并命名为 Example6-5.html。

在 Example6-5.html 中输入文字,操作过程和 6.5.1 节中一样,为文字选择“.font”样式,在浏览器中显示出来的效果和图 6.71 一样。



图 6.72 CSS 样式模板

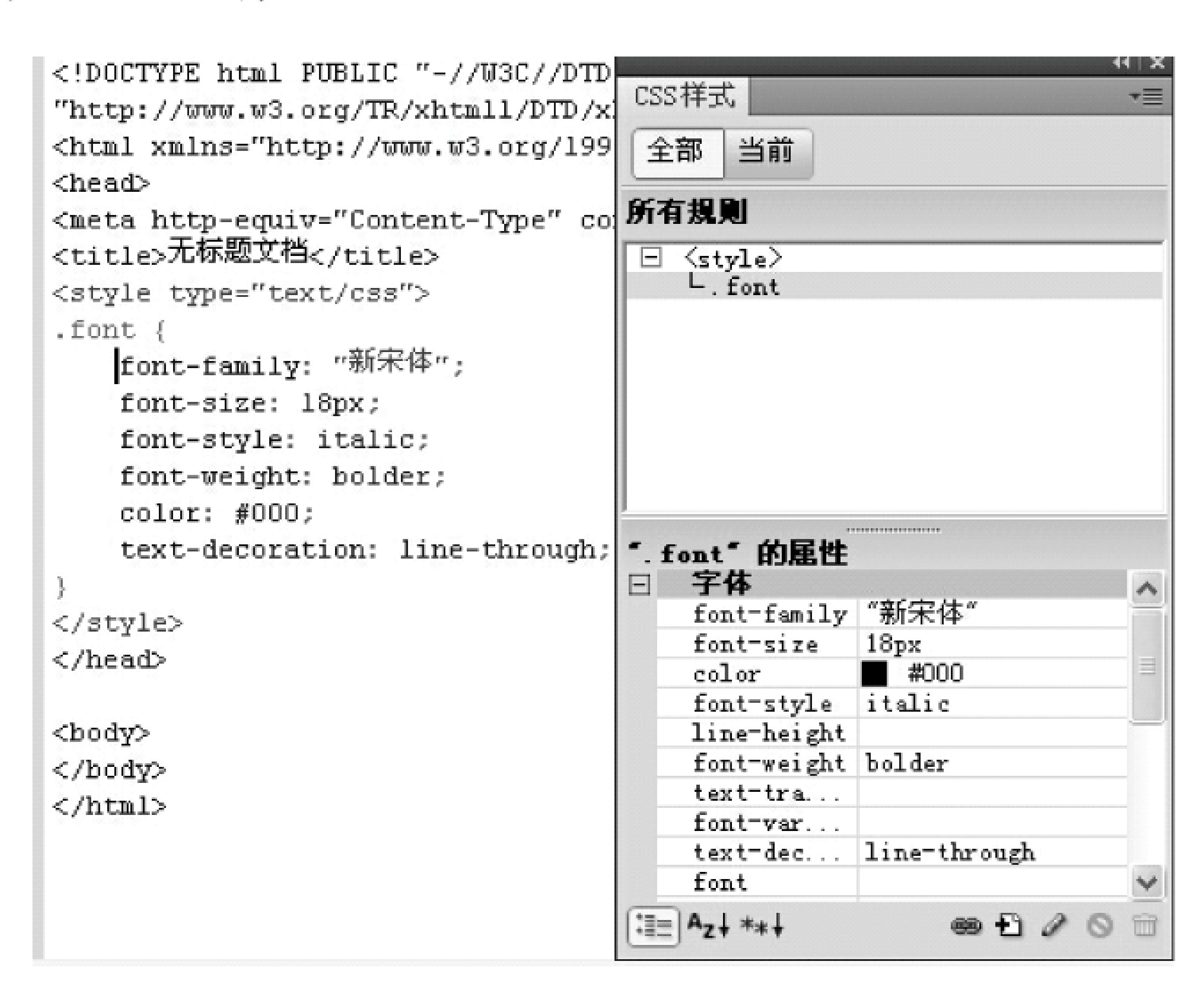


图 6.73 网页中的 CSS 样式

小结

本章从 Dreamweaver 的基础知识入手,介绍了 Dreamweaver 的工作界面的组成及页面属性的设置。讲解了页面内容的插入与使用方法、网页排版和布局,以及 CSS 的使用。通过本章的学习,希望读者能初步掌握简单网页制作的基础知识,为进一步的学习打下基础。

习题

1. 段落对齐有左对齐、右对齐、________和________ 4 种方式。
2. 在 Dreamweaver 中插入图像的方法主要有________和________两种。
3. 大多数浏览器默认情况按边框粗细设置为________显示表格。
4. 如何在 Web 页面中添加多个空格?
5. 框架的创建有哪几种方法?

6．设置表格或单元格宽度时用到的像素和百分比有何区别?

7．根据本章所学内容,请读者自己用表格创建一个 Web 页面,在文档中输入自己的简历,并根据自己的想法添加字体列表,并设置字体的大小、颜色及段落对齐方式,并在页面内插入一些图片。

8．对比 Example6-4.html 和 Example6-5.html 代码,用手动输入代码的方法完成这两个例子,并去掉删除线,字体设定为红色。

9．根据本章所学内容,请读者自己创建一个基于框架的网站。

第7章 动画制作工具Flash

7.1 Flash 简介

Flash 是 Macromedia 公司开发的一款优秀的网页动画制作软件，它广泛应用于网络中的多种领域，从简单的动画到复杂的交互式 Web 应用程序，无所不在。用户在友好的创作环境下通过添加图片、声音和视频等对象，就可以制作出精美绚丽的 Flash 动画。Flash 和 Dreamweaver、FireWorks 并称为“网页三剑客”。

Flash 是 Web 设计人员、交互式媒体专业人员进行设计的理想著作工具。该工具注重创建、导入和处理多种类型的媒体（包括音频、视频、位图、矢量、文本和数据），提供了对 Web 团队（由设计人员和开发人员组成）成员之间的工作流程进行优化的项目管理工具。外部脚本撰写和处理数据库中动态数据的能力及其他功能使得 Flash 特别适用于大规模的复杂项目。

7.1.1 Flash 的特点

1. 优化的操作界面

Flash 的操作界面经过重新设计，界面更美观，层次更清晰，各面板布局更合理。相对于其他制作动作的程序，Flash 更加容易操作，无须任何编程基础就可以轻松制作出动画效果。

2. 生成的动画文件可以独立播放

用 Flash 制作的动画作品不仅可以在线观看，也可以离线观看，并保留其原来动画中的各种交互式操作功能。

3. 流媒体动画

Flash 播放器在下载 Flash 影片时采用流媒体方式，也就是说，在 Flash 文件还没有完全下载完毕时就播放动画，即下载的同时进行播放。

4. 文件体积较小

由于 Flash 采用的图像方式是矢量图，生成的文件相对于传统网页动画中同等体积的

像素图来说要小很多。

5. 具有交互式作用的多媒体影片

Flash 影片可以通过 ActionScript 脚本语言与使用者进行交互式联系，使用者可以通过键盘操作或鼠标操作与影片实现互动。

6. 易用性

由于其操作界面与其他程序(如 Office、Photoshop 等)相似，因此用户很容易上手。

7.1.2　Flash 的启动界面

打开 Flash CS6 软件，其欢迎界面显示如图 7.1 所示，下面按区域介绍各部分功能。



图 7.1　Flash 启动界面

1. 从模板创建

在该选项区的列表中单击相应的模板类别选项，即可弹出“从模板创建”对话框，并自动切换到所选的模板类别。

2. 打开最近的项目

在该选项区中显示了最近打开过的 Flash 文档，单击相应的文档，即可快速在 Flash CS6 中打开该文档。如果单击“更多”选项，则会弹出“打开”对话框，可以在该对话框中浏览到需要打开的 Flash 文档。

3. 相关资源

在该选项区中提供了 Flash CS6 相关资源的快速访问链接，单击相应的选项即可在浏览器窗口中打开 Adobe 官方网站所提供的相关内容介绍。

4. 新建

在该选项区的列表中提供了 Flash CS6 所支持的所有文档类型，单击相应的文档类型即可自动创建默认设置的该类型文档。

5. 扩展

在该选项区中提供了 Flash CS6 的扩展选项，单击 Flash Exchange 选项，将自动在浏览器窗口中打开 Adobe 官方网站的软件扩展页面。

6. 学习

在该选项区中提供了 Flash CS6 相关功能的学习资源，单击相应的选项即可在浏览器窗口中打开 Adobe 官方网站所提供的相关的内容介绍页面。

一般初学者可以在新建中选择创建 fla 文件，一个 fla 文件就表示一个 Flash 源文件，在欢迎界面中选择新建“Flash 文件(ActionScript 3.0)”。

7.1.3 Flash CS6 的工作区

Flash 的用户界面由几个主要部分组成，最上方的是主菜单栏。它分类提供了 Flash CS6 中所有的操作命令。时间轴和舞台位于工作界面的中心位置。浮动面板和工具箱位于工作界面的最右边。在工具箱中提供了 Flash 中所有的操作工具，如笔触颜色和填充颜色，以及工具的相应设置选项，通过这些工具可以在 Flash 中进行绘图、调整等相应的操作，如图 7.2 所示。

1. 菜单栏

在菜单栏中分类提供了 Flash CS6 中所有的操作命令，几乎所有的可执行命令都可在这里直接或间接地找到相应的操作选项。

(1) “文件”菜单

“文件”菜单下的菜单命令多是具有全局性的，如“新建”、“打开”、“关闭”、“保存”、“导入”、“导出”命令、“发布”、“AIR”设置、“ActionScript 设置”、“打印”、“页面设置”以及“退出”等命令。

(2) “编辑”菜单

在“编辑”菜单中提供了多种作用于舞台中各种元素的命令，如“复制”、“粘贴”、“剪切”等。另外在该菜单下还提供了“首选参数”、“自定义工具面板”、“字体映射”及“快捷键”的设置命令。

(3) “视图”菜单

在“视图”菜单中提供了用于调整 Flash 整个编辑环境的视图命令，如“放大”、“缩小”、

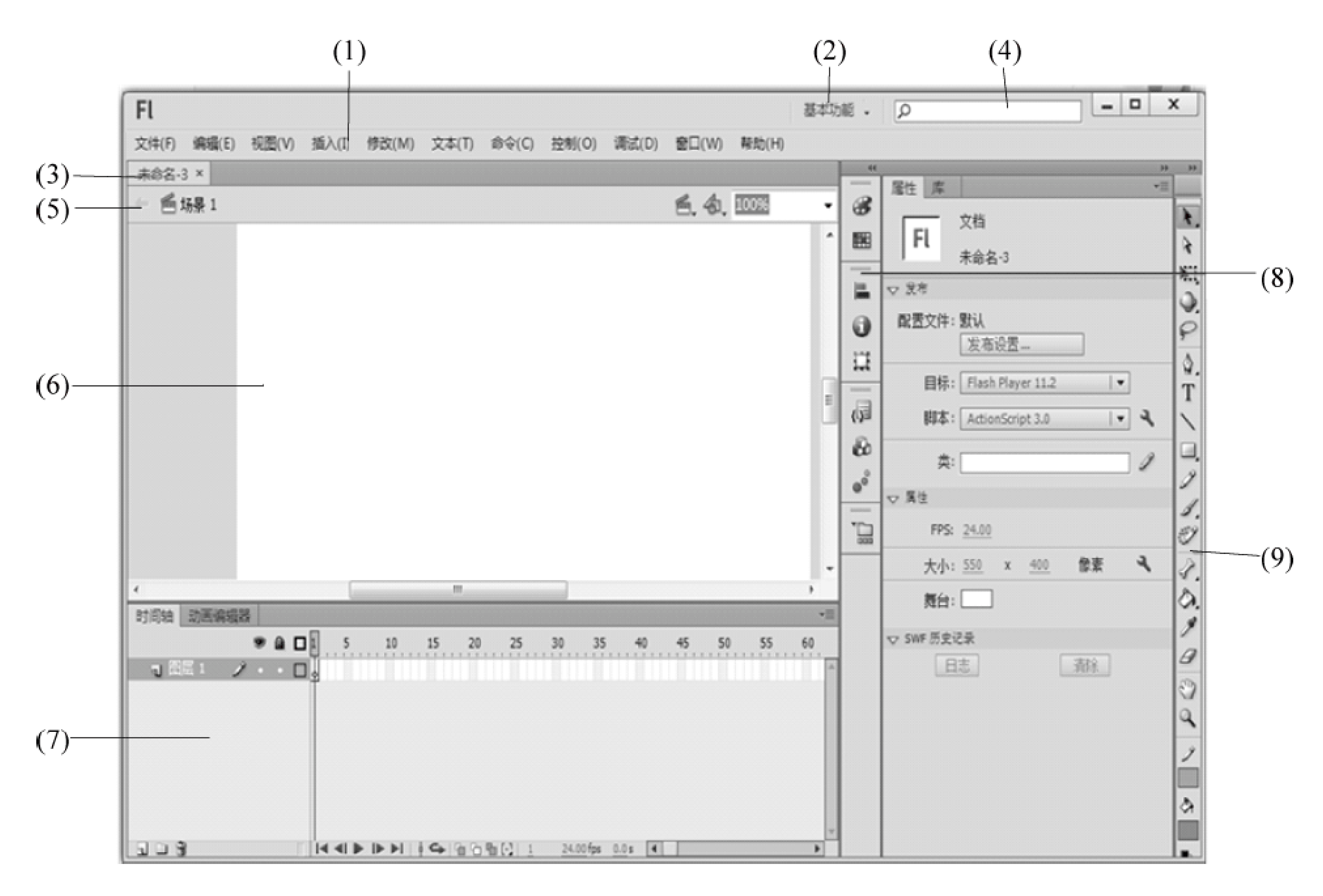


图 7.2　Flash CS6 编辑界面

“标尺”、“网格”等命令。

(4)“插入”菜单

在“插入”菜单中提供了针对整个“文档”的操作，比如在文档中“新建元件”、“场景”，在“时间轴”中插入“补间”、“层”或“帧”等。

(5)“修改”菜单

在“修改”菜单中包括了一系列对舞台中元素的修改命令，如“转换为元件”、“变形”等，还包括了对文档的修改等命令。

(6)“文本”菜单

在“文本”菜单中可以执行与文本相关的命令，如设置“字体”、“样式”、“大小”、“字母间距”等。

(7)“命令”菜单

Flash CS6 允许用户使用 JSFL 文件创建自己的命令，在“命令”菜单中可运行、管理这些命令或使用 Flash 默认提供的命令。

(8)“控制”菜单

在“控制”菜单中可以选择“测试影片”或“测试场景”，还可以设置影片测试的环境，例如：用户可以选择在桌面或移动设备中测试影片。

(9)“调试”菜单

在“调试”菜单中提供了影片调试的相关命令，如设置影片调试的环境等。

(10)“窗口”菜单

在“窗口”菜单中主要集合了 Flash 中的面板激活命令，选择一个要激活的面板的名称即可打开该面板。

(11) “帮助”菜单

在“帮助”菜单中含有 Flash 官方帮助文档，也可以选择“关于 Adobe Flash Professional”来了解当前 Flash 的版权信息。

2. 基本功能

Flash CS6 提供了多种软件工作区预设，在该选项的下拉列表中可以选择相应的工作区预设，选择不同的选项，即可将 Flash CS6 的工作区更改为所选择的工作区预设。在列表的最后提供了“重置基本功能”、“新建工作区”、“管理工作区”3 种功能，“重置基本功能”用于恢复工作区的默认状态，“新建工作区”用于创建个人喜好的工作区配置，“管理工作区”用于管理个人创建的工作区配置，并可执行重命名或删除操作，如图 7.3 所示。



图 7.3 下拉列表

3. “文档窗口”选项卡

在“文档窗口”选项卡中可显示文档名称，当用户对文档进行修改而未保存时，则会显示“ * ”号作为标记。如果在 Flash CS6 软件中同时打开了多个 Flash 文档，可以单击相应的文档窗口选项卡进行切换。

4. 搜索框

该选项提供了对 Flash 中功能选项的搜索功能，在该文本框中输入需要搜索的内容，再按 Enter 键即可。

5. 编辑栏

左侧显示当前“场景”或“元件”，单击右侧的“编辑场景”按钮，在弹出的菜单中可以选择要编辑的场景。单击旁边的“编辑元件”按钮，在弹出的菜单中可以选择要切换编辑的元件。如果希望在 Flash 工作界面中设置显示/隐藏该栏，则可以执行“窗口”→“工具栏”→“编辑栏”命令，即可在 Flash CS6 工作界面中设置显示/隐藏该栏。

6. 舞台

舞台是用户在创建 Flash 文件时放置图形内容的区域，这些图形内容包括矢量插图、文本框、按钮、导入的位置或者视频等。如果需要在舞台中定位项目，可以借助网格、辅助线和标尺。Flash 工作界面中的舞台相当于 Flash Player 或 Web 浏览器窗口中在播放 Flash 动画时显示 Flash 文件的矩形空间。

7. “时间轴”面板

时间轴用于组织和控制文档内容在一定时间内播放的图层数和帧数。与胶片一样，Flash 文件也将时长分为帧。图层就像是堆叠在一起的多张幻灯片，每个图层都包含一个显示在舞台中的不同图像。时间轴的主要组件就是图层、帧和播放头。

8. 浮动面板

浮动面板用于配合场景、元件的编辑和 Flash 的功能设置，在“窗口”菜单中执行相应的命令，可以在 Flash CS6 的工作界面中显示/隐藏相应的面板。

9. 工具箱

在工具箱中提供了 Flash 中所有的操作工具，如笔触颜色和填充颜色，以及工具的相应设置选项，通过这些工具可以在 Flash 中进行绘图、调整等相应的操作，如图 7.4 所示。

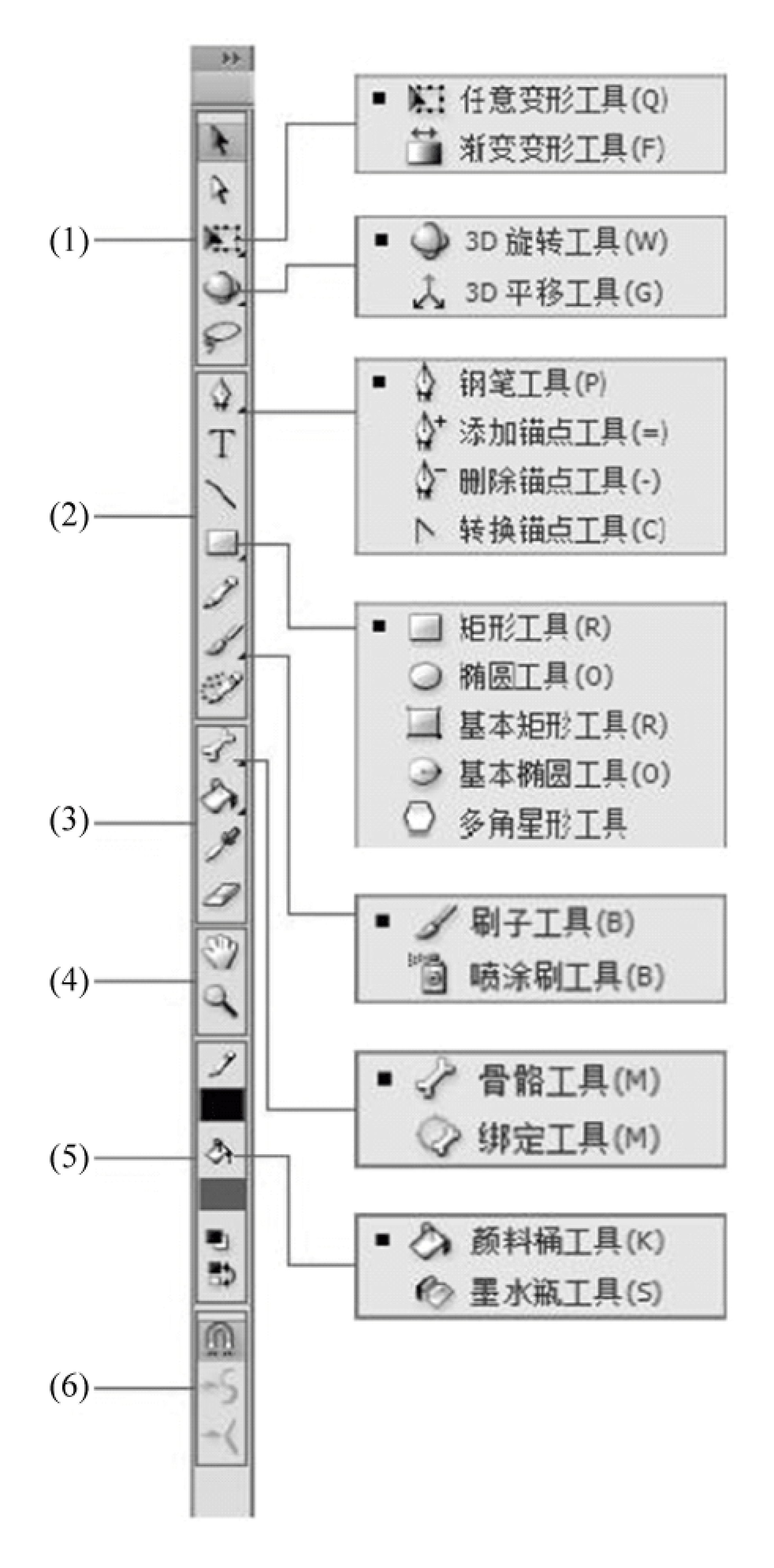


图 7.4　工具箱

(1) 选择变换工具

工具箱中的选择变换工具包括了“部分选择工具”、“套索工具”、“任意变形工具”和“渐变变形工具”，利用这些工具可对舞台中的元素进行选择、变换等操作。

(2) 绘画工具

绘画工具包括“钢笔工具组”、“文本工具”、“线条工具”、“矩形工具组”、“铅笔工具”、“刷子工具组”以及“Deco 工具”，这些工具的组合使用能让设计者更方便地绘制出理想的作品。

(3) 绘画调整工具

该组工具能让设计者对所绘制的图形、元件的颜色等进行调整，它包括“骨骼工具组”、“颜料桶工具组”、“滴管工具”、“橡皮擦工具”。

(4) 视图工具

视图工具中含有“手形工具”用于调整视图区域，“缩放工具”用于放大/缩小舞台大小。

(5) 颜色工具

颜色工具主要用于“笔触颜色”和“填充颜色”的设置和切换。

(6) 工具选项区

工具选项区是动态区域，它会随着用户选择的工具的不同而显示不同的选项，如果单击工具箱中的“套索工具”按钮，单击“魔术棒”按钮，则切换“套索工具”为“魔术棒工具”，单击“魔术棒设置”按钮，用于设置“魔术棒”的相关参数。

7.2 Flash 的基本概念与操作

7.2.1 Flash 的基本概念

1. 帧

在 Flash 中，帧是进行 flash 动画制作的最基本单位，它装载了 Flash 播放的内容。在时间轴上的每一帧都可以包含需要显示的所有内容，包括图形、声音、各种素材和其他多种对象。

时间轴中有很多单元格，每一个横行代表一个层，层内的每一个小单元格表示一帧。帧是创建动画的基本单位。在 Flash 中包含有几种类型的帧，它们是：关键帧、空白关键帧和过渡帧。

(1) 空白关键帧

空白关键帧是关键帧的一种，它什么内容都没有。启动 Flash 后，在默认状态下，每个层的第一帧都是空白关键帧。在动画制作工作区中添加了内容后，它自动变成关键帧。

默认的新建文件会含有一个空白关键帧，空白关键帧在时间轴上显示为没有标志的空心单元格，如图 7.5 所示。

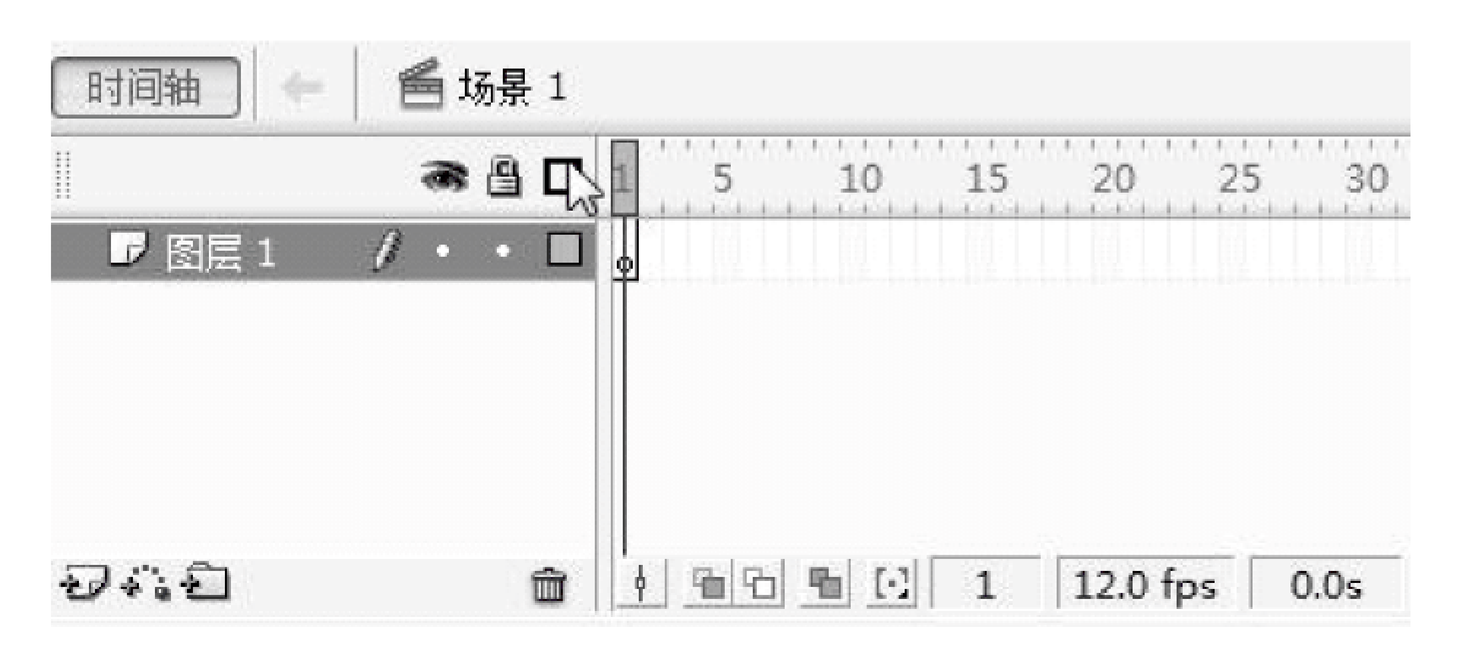


图 7.5 空白关键帧

(2) 关键帧

关键帧是决定一段动画的必要帧,它含有关键内容。在其中可以放置图形,播放对象,并可以对所包含的内容进行编辑。关键帧一般放在动画开始点、控制转折点或结束点。在空白关键帧中添加了内容后就变成了关键帧,在时间轴中包含内容的关键帧显示为有黑色实心圆点的单元格,如图 7.6 所示。

(3) 过渡帧

在两个关键帧中间的普通帧称为过渡帧。在 Flash 中,确定了两段的关键帧后,利用命令可以自动计算添加过渡帧,无须人工添加。

过渡帧的显示状态与过渡方式有关,如果是运动渐变过渡,过渡帧就显示为带有箭头直线段的浅紫色单元格,如图 7.7 所示。

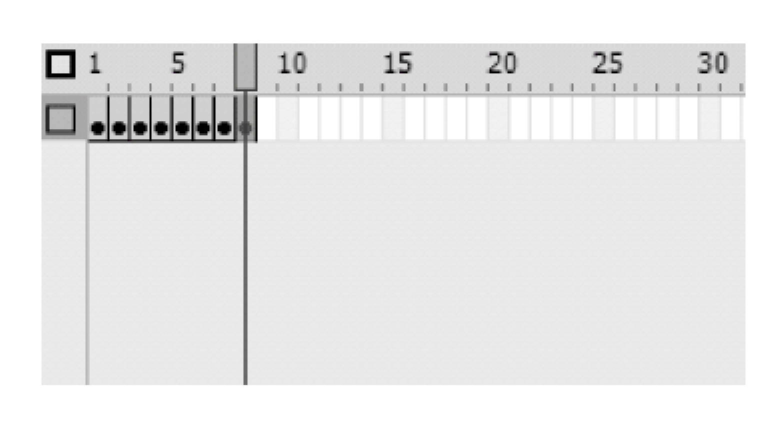


图 7.6 关键帧

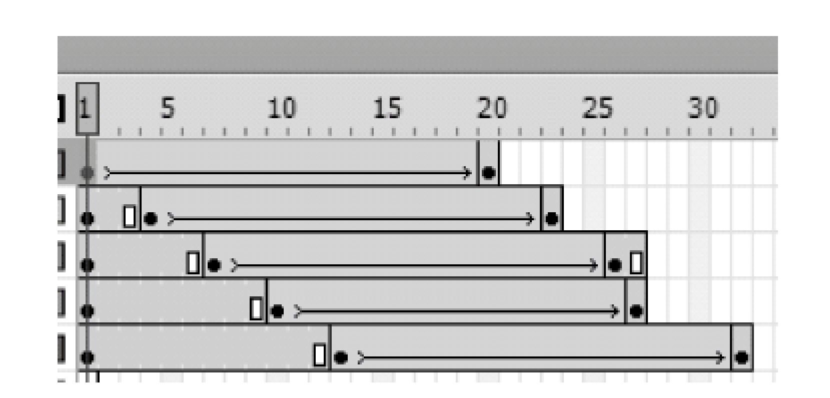


图 7.7 运动渐变过渡

如果是形状渐变过渡,过渡帧就显示为带有箭头直线段的浅绿色单元格,如图 7.8 所示。

如果没有按照要求准备好过渡条件就过渡,就会出现错误,出错时的过渡帧如图 7.9 所示,是虚线的单元格。

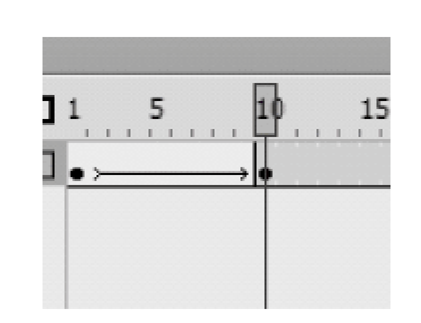


图 7.8 形状渐变过渡

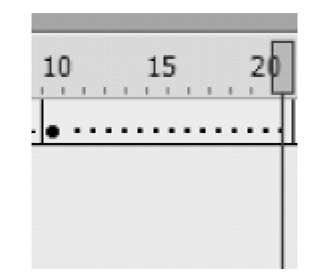


图 7.9 出错

2. 层

与 Photoshop 相同,在 Flash 中,层也好比一张"透明纸"。首先需要在一张透明的纸上画一些图形,写一些文字,然后将多层"透明纸"按一定的顺序进行叠加,就实现最终的效果了,层在时间轴窗口中就是一个横行,它包含着若干个帧。

Flash 软件的图层位于时间轴面板的左侧,其结构如图 7.10 示。在最顶层的对象将始终显示于最上方,图层的排列顺序决定了舞台中对象的显示情况。在舞台中每个层的对象可以设置任意数量,如果时间轴面板中图层的数量过多的话,可以通过上下拖动右侧的滑动条观察被隐藏的图层。

在 Flash 中,分为普通层、运动引导层、遮罩层和注释说明层 4 种层。通常建立的层都是普通层;通过运动引导建立的层是引导层,它的作用是提供引导线作为被引导层中对象的运动轨迹;遮罩层是通过设定遮罩关系建立的层,它的作用是为了实现遮罩关系下的特别效果;在注释说明层中可以添加一些说明性文字,输出时不输出该层内容。

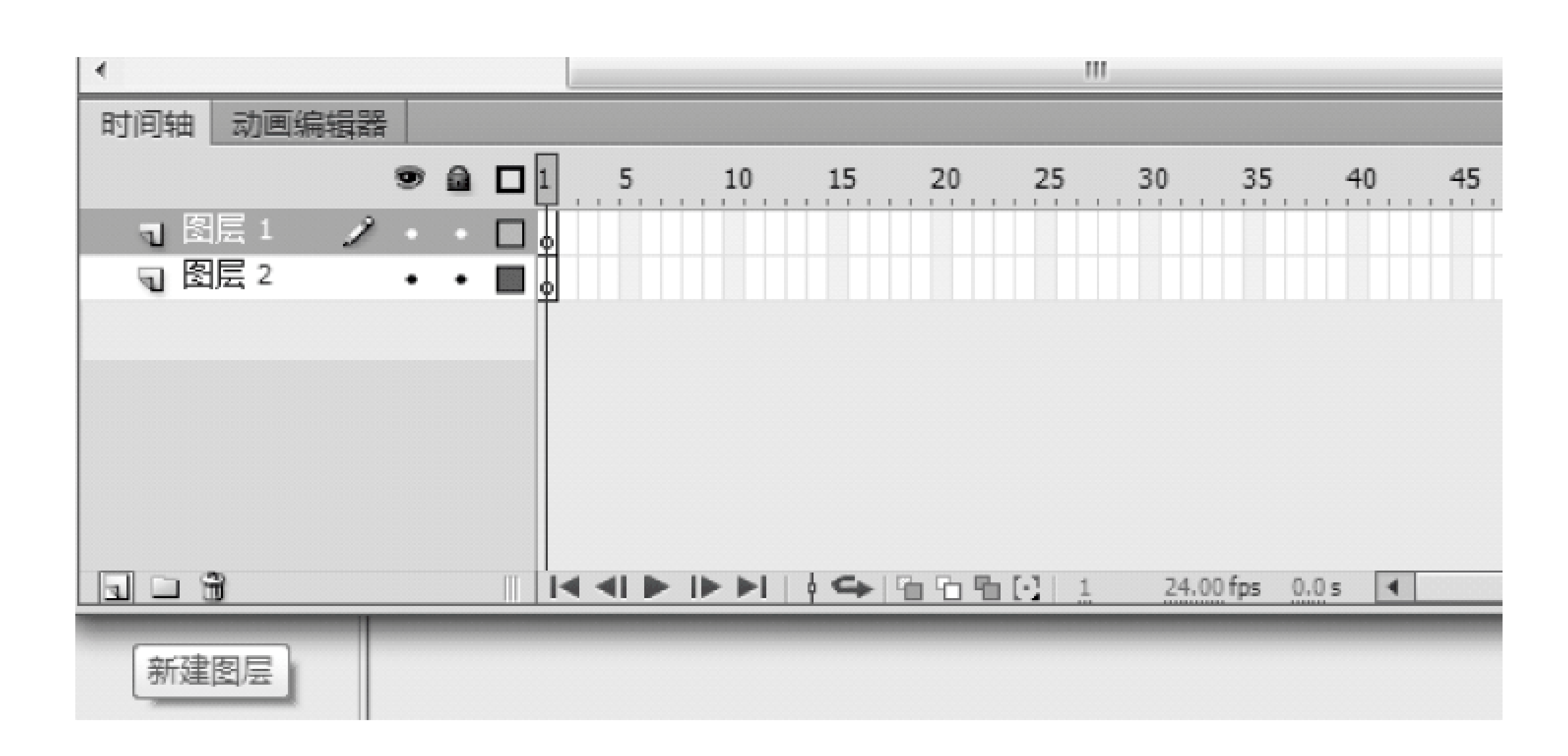


图 7.10　图层

3. 对象

在 Flash 中任何被选中的东西，都可以叫做对象。对象被选中后，可以通过设置实现用户要求的变形或运动。如果绘制了一个圆，并选中这个圆的话，它就是一个对象，可以通过工具对它进行填充、修改、移动、变形等编辑。

用户通过工具箱中的工具直接绘制的对象称为独立对象；如果用户选中了这些对象，利用“组合”命令形成了一个新的对象，称为组合对象；其中的各个独立对象组成群组对象的基本对象，称为子对象。

4. 元件、实例和库

元件也称为符号或图符，就是将某段 Flash 剪辑、导入的视频、动画或图片等作为一个整体存放在元件库中，事后可将其拖放到舞台被使用。Flash 中包含 3 种元件类型，分别为图形元件、影片剪辑元件和按钮元件。元件是 Flash 动画的重要组成部分。元件的主要特性是可以被重复利用。Flash 中的所有元件都被归纳在库面板中，可以被随时调用，十分方便。即便在场景中将所有元件全部删除掉，也不会影响库面板中的元件。每个元件都具有独立的时间轴、工作区和层，除此之外，每种元件类型都具有独特的属性，只有了解三种元件类型的特性，才能使元件的作用得到充分的发挥。

创建元件一般有两种方法，一种方法是将舞台上的选定对象转换为元件。另一种方法是创建一个空白元件，然后绘制或导入需要的对象。元件的创建方法在后面的实例中再详细说明。

在 Flash 中创建的所有元件都会出现在库面板中，拖曳库面板中的元件到场景中，可以反复利用元件。应用于影片的元件被称做“实例”。在工作区中选中实例，通过元件属性面板中的设置，可以改变实例的尺寸、颜色与元件类型，但这些改动只针对实例本身，不会影响到库面板中的元件。

Flash 文档的库中存储了用户创建或导入的媒体资源。库是 Flash 组织、编辑和管理动画中创建和使用的各种元件的仓库。所有元件都会被自动载入到当前影片的库面板中，以便以后灵活调用。另外，还可以从其他影片的库面板中调用元件，或者根据需要建立自己的库。

7.2.2　动画的基本概念

1. 计算机动画的概念与原理

所谓动画，简单地说，就是将静止状态下的一幅幅图画或照片连续地演播出来的技术，具体来说又有以下两种说法：

(1) 动画是一种通过一系列连续的画面来显示运动的技术，是以一定的播放速度来达到连续运动的效果的；

(2) 动画是一系列具有各种物体的动态画面的处理过程，每一个画面都可以与前一个画面有着微小的变化。

这两种不同的说法，从不同的角度阐述了动画的一些特性——连续性、运动性，以及是由有微小差别的多个画面组成的等特性。

动画的概念已经明确，原理就自然清楚了——人的“视觉原理”。医学证明，人类的视觉系统有视觉暂留的特性，也就是说，人的眼睛看到图像后的 1/24s 内，图像不会从大脑中消失。利用这一原理，如果在前一图像没有消失之前，播放下一图像，就会给人造成一种连续变化的视觉效果。电影、电视都是利用了这一原理。

2. 计算机动画的分类

近年来，涌现出许多计算机动画系统，它们的方式、目的、特点各不相同。对这些系统进行大致的分类如下。

(1) 按计算机在动画制作中的作用分类

按计算机在动画制作中的作用分类，可将计算机动画系统分为两类：计算机辅助动画和造型动画。前者属于二维动画，其主要用途是辅助动画师们制作类似于传统动画的动画，以及对各类动画片段的编辑(合成、连接)等后期制作。后者属于三维动画，用来绘制和控制在三维空间中运动的物体。

(2) 按运动的控制方式分类

根据运动的控制方式不同，可将计算机动画分为关键帧动画和算法动画。关键帧动画通过对关键画面的控制来获得中间画面的动画序列，算法动画则采用算法实现对物体的运动控制或模拟摄像机的运动控制。

(3) 从制作原理上分类

从制作原理上分类，动画可简单划分为两大类：逐帧动画和补间动画。

逐帧动画就是通过更改每一帧中的舞台内容而获得动画效果，它最适合于每一帧中的图像都在改变，而不是仅仅简单地在舞台中移动的复杂动画。逐帧动画的缺点是太耗费时间和精力，而且最终生成的动画文件偏大。但是，它也有自己的优点，即能最大限度地控制动画的变化细节。

补间动画是在两个关键帧之间建立渐变的一种动画。补间动画关键帧的对象是元件的实例、群组体或文字。补间动画的原理是：在第一个关键帧中设置元件实例、群组体或文字的属性，然后在第二个关键帧中修改对象的属性，从而在两帧之间产生动画效果。可以修改的属性包括大小、颜色、旋转和倾斜、位置、透明度及各种属性的组合。

3. Flash 动画的格式

在制作 Flash 动画之前，首先了解一下 Flash 动画的几种常见的格式。

（1）Flash 源文件格式

Flash 源文件格式的扩展名为.fla，该类型文件只能在 Flash 应用程序中被打开和编辑，在其他软件中是无法打开的。用户可以在 Flash 应用程序中打开该文件，然后导出扩展名为 swf 或 swt 的文件，以便在浏览器中使用。

（2）Flash 电影文件格式

Flash 电影文件格式的扩展名为 swf。该类型文件是由电影文件格式源文件压缩生成的文件，为方便在网络中查看，已对其进行了优化处理。因此，这种类型的文件可以在浏览器中播放，但是要求浏览器安装 Flash Player 播放器插件后才能播放，也可以在 Dreamweaver 中预览，但是不能在 Flash 应用程序中编辑。另外，在 Dreamweaver 中允许用户直接插入 Flash 动画对象，该对象的扩展名为 swf。

7.3 动画制作

Flash 提供了两种创建动画的方式，它们是逐帧动画与补间动画。其中，补间动画又可以分为运动渐变动画和形状渐变动画两种。制作动画时，用户可以轻松地使对象在场景中来回地运动，在运动的过程中还可以伴随有诸如放大、缩小、旋转、变色及淡入淡出等丰富效果。

7.3.1 逐帧动画的制作

逐帧动画的制作方法与传统的动画相似，将对象的运动过程分解成多个静态图形，再将这些连续的静态图形置于连续的关键帧中，就构成了逐帧动画。

由于其中的每一帧都是关键帧，所以生成的影片相对较大，仅适用于制作较为精细复杂的小动画。

例 7.1 制作小马快跑的动画效果。

本实例需要准备 8 个大小相同的图像文件，文件名分别是 1.gif，2.gif，…，8.gif，如图 7.11 所示。

制作过程如下。

（1）创建影片文档

执行“文件”→“新建”命令，在弹出的对话框中选择“Flash 文档”选项后，单击“确定”按钮，新建一个影片文档。在“文档属性”对话框中进行设置，文件大小为 150×170 像素，背景色为白色，如图 7.12 所示。

（2）修改图层名称

双击“图层 1”的图层名称，将其修改为“小马”，如图 7.13 所示。

（3）导入小马图片

单击此层第一帧，执行“文件”→“导入”→“导入到舞台”命令，弹出如图 7.14 所示的对

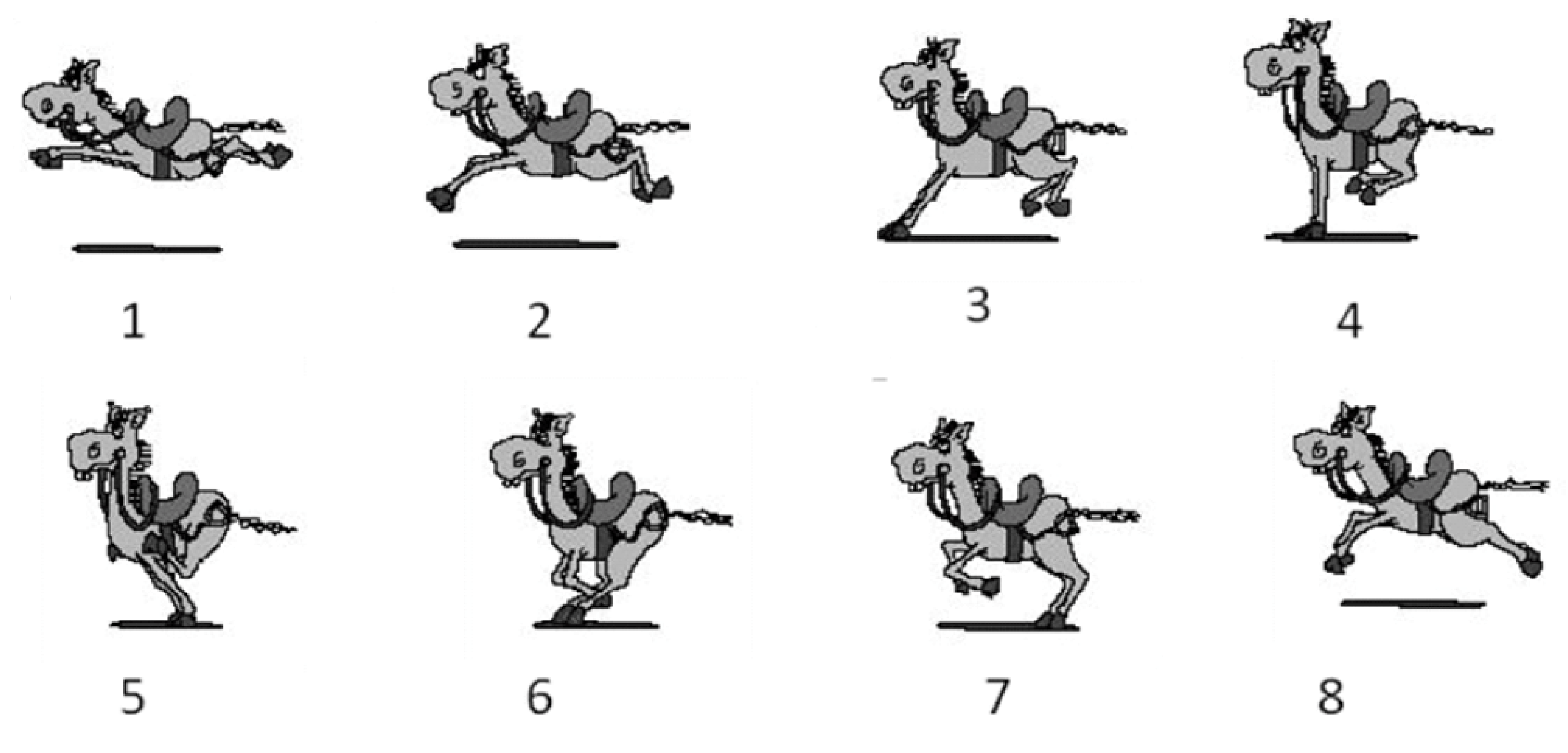


图 7.11　逐帧动画制作

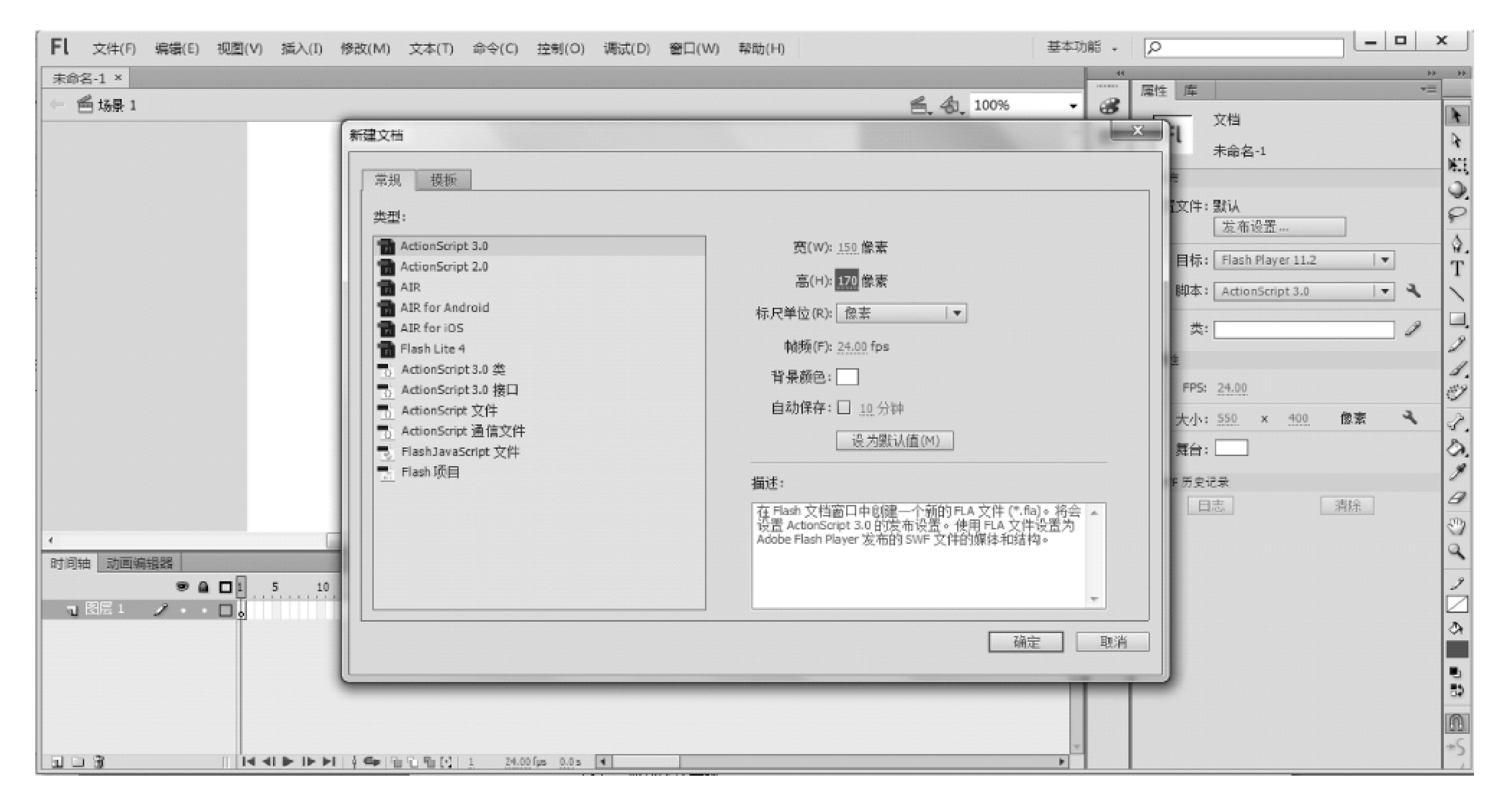


图 7.12　创建影片文档

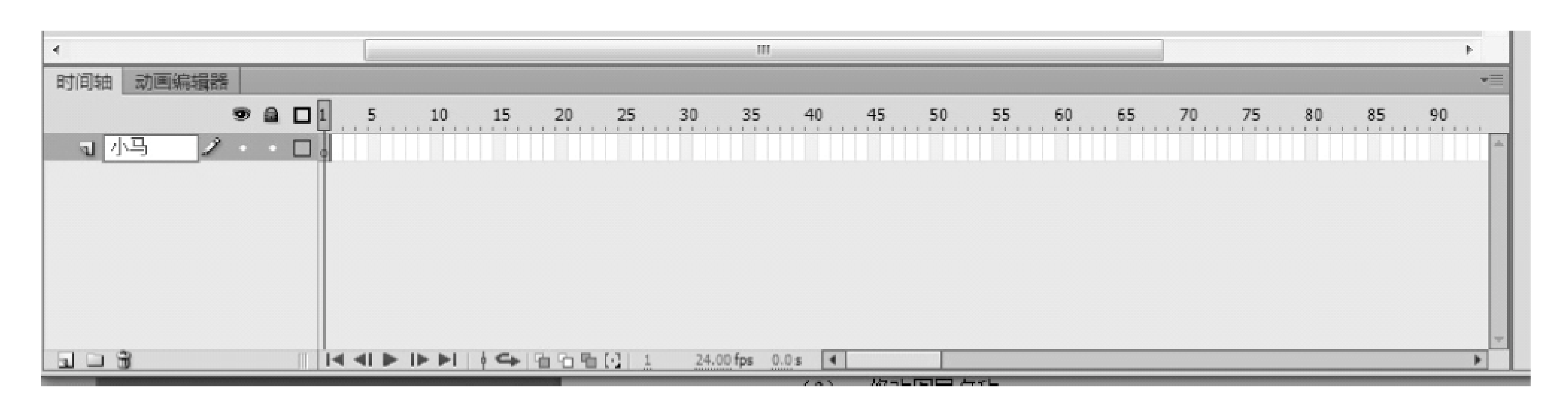


图 7.13　修改图层名称

话框，将素材中的系列小马图片导入。

在图 7.15 选择“是”按钮，Flash 会自动把该文件夹中的图片按序号导入到场景中。导入后，动画序列 Flash 自动分配在 8 个关键帧中，如图 7.16 所示。

也可以手动插入图片，在第 8 帧处单击鼠标右键选择插入关键帧，再将需要插入的图片



图 7.14 导入小马图片

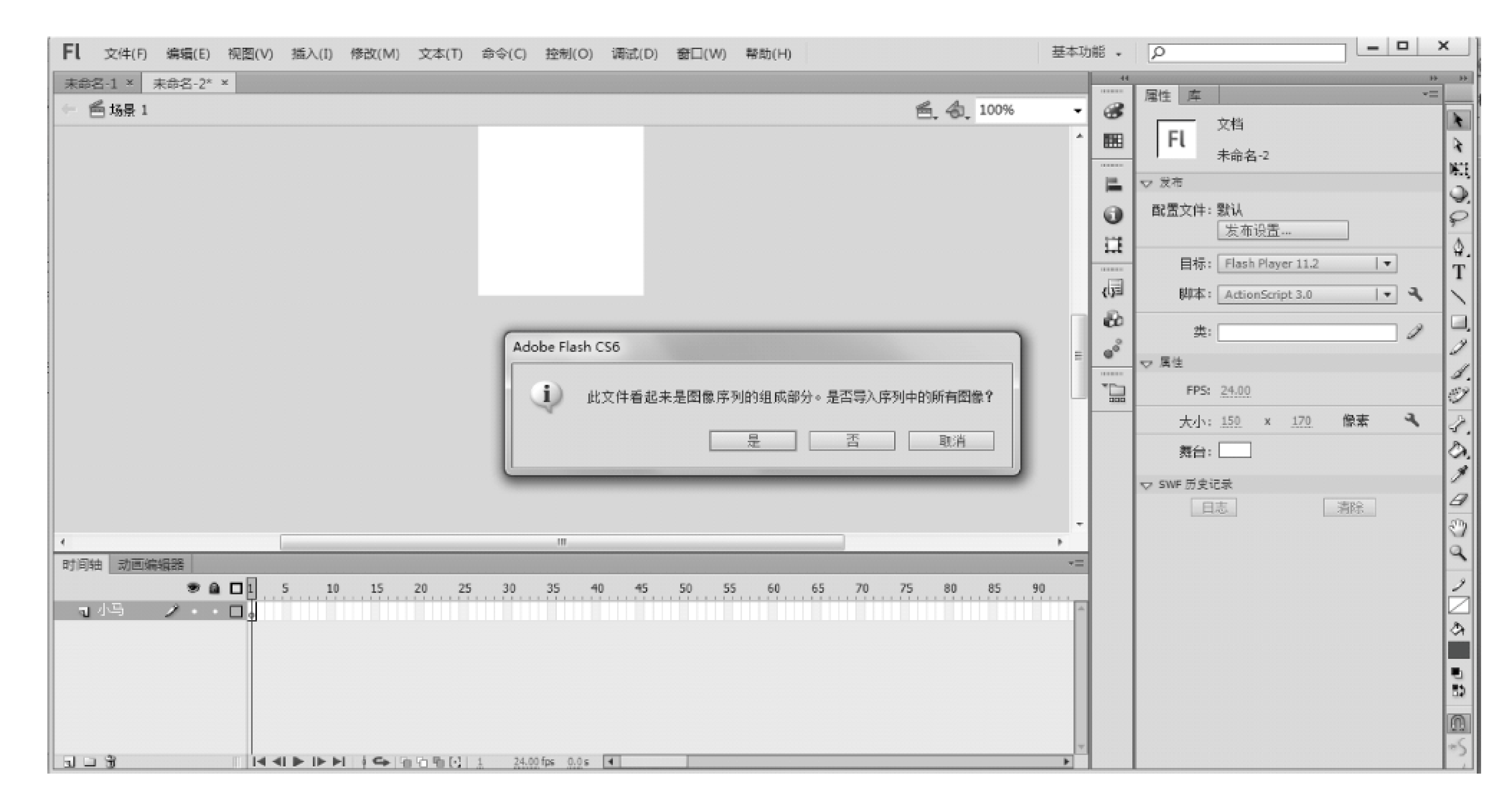


图 7.15 导入图片

导入至舞台，在第 8 帧处单击鼠标右键在选项卡中选择“插入关键帧”即可。

(4) 测试存盘

执行“控制”→“测试影片”命令，观察动画效果。如果满意，执行“文件”→“保存”命令，将文件保存成 horse.fla 文件。如果要导出 Flash 播放文件，执行“文件”→“导出”→“导出影片”命令即可。

7.3.2 运动渐变动画的制作

运动渐变动画是补间动画的一种。通过为对象创建运动渐变，可以改变对象的位置、大小、旋转或倾斜，做出物体运动的各种效果。通过设置实例的颜色属性，还可以制作出丰富的渐变效果，例如，实例的淡入淡出效果。

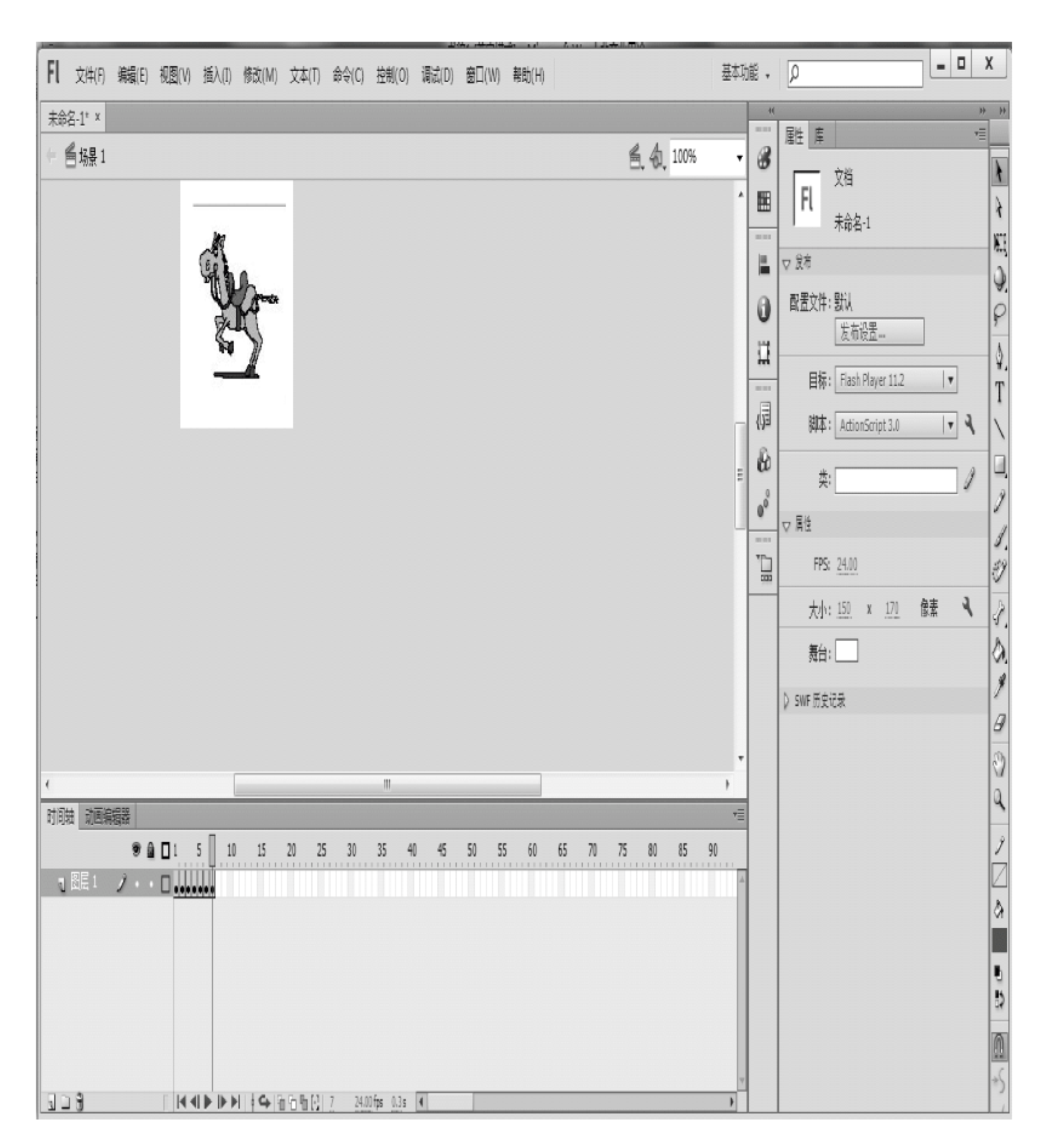


图 7.16　小马系列图片导入

要正确运用运动渐变必须满足以下创作条件：

(1) 运动渐变只能作用于元件实例，要对形状、组合、位图或文本对象应用运动渐变，必须先将这些对象转换为元件；

(2) 运动渐变中的元件只能是一对一的渐变，多对一或一对多均会导致渐变失败；

(3) 运动渐变中的元件只能在同一图层中。

例 7.2　使用运动渐变动画实现文字被风吹走的效果。

制作过程如下。

(1) 创建影片文档

执行"文件"→"新建"命令，在弹出的对话框中选择"Flash 文档"选项后，单击"确定"按钮，新建一个影片文档。在"文档属性"对话框中进行设置：文件大小为 550×350 像素，背景颜色为＃3333cc。

(2) 创建元件

在制作本实例动画之前，应当先掌握元件和库的基本操作。

创建元件的方法主要两种方法：

第一种方法是直接创建一个空白元件，然后自动进入元件编辑模式，接着在元件中创建和编辑元件中的对象。单击菜单栏中的"插入"选择"新建元件"，再在对话框中设置元件的

名称，选择元件的类型，最后单击“确定”按钮。

第二种方式是将对象转换为元件。将选定的对象转换为元件，右击选定的对象，弹出快捷菜单。对象可以为任何元素，将对象转换为元件后，即可改变对象的属性。

创建的元件会自动加入到库中，执行“窗口”→“库”命令或按 Ctrl+L 快捷键，即可打开库面板，如图 7.17 所示。



图 7.17 库

库面的基本操作包括以下 4 种。

① 新建元件

单击库面板左下角的“新建元件”按钮，即可开始新建元件。

② 新建文件夹

为了便于管理，经常把一批相关的元件放在一个组内。单击“新建文件夹”按钮，出现新文件夹，填入文件夹名称，确定后完成新建文件夹。

③ 属性的修改

选中库面板中的元件，单击“属性”按钮，即可修改和调整元件的属性。

④ 删除元件

选中库面板中的元件或文件夹，单击“删除”按钮，弹出“删除”对话框，提示删除后无法撤销。如果单击“删除”按钮，即可实现永久删除。

在本实例中，因为每一个字都要单独做动画，所以必须把每个字单独做成一个元件，然后分别放在不同的层上。

选择文字工具，在属性面板中设置文字的字体为华文楷体，大小为 80，颜色为 #cc6699，在场景中输入“欢”，然后选中输入的“欢”字，按 F8 键将其转换为元件或在“欢”字上单击鼠标右键，选择“转换为元件”。

按照类似的方法，分别输入“欢”、“迎”、“学”、“习”、Flash，并分别将它们转化为元件。

(3) 将元件分布到各个图层中并设置对其方式

元件制作完毕后，在时间轴窗口新建 5 个图层，使层的总数变为 5 个。按 Ctrl+L 键打开库面板，在最上面一层，把“欢”字拖到主场景中，并改变图层的名字为“欢”。在第二层把“迎”字拖到主场景中，并改变图层的名字为“迎”。其余类推，把这 5 个字分别放在不同的层上，注意，在最下面的图层 Flash 中只保留文字 Flash 的元件，将其他元件删除即可，如图 7.18 所示。

然后用箭头工具框选所有文字，按 Ctrl+K 键打开对齐面板，设置对象的“对齐”为垂直方向底对齐，“分布”为水平居中分布，“相对于舞台”为相对于舞台分布。

（4）设置文字的缩放和颜色变化

本实例中的文字变化是通过改变实例颜色样式的方法实现的。设置如下：利用属性面板中“色彩效果”选项的设置，改变实例的颜色，如图 7.19 所示。



图 7.18　各个图层



图 7.19　色彩效果

在工作区中选中元件的实例，从“色彩效果”下拉列表中进行选择，其中包括以下 5 种颜色样式。

① 无：默认设置，表示不为实例设置颜色效果。

② 亮度：用于设置实例的亮度。

③ 色调：用于设置实例的色彩倾向。

④ Alpha：用于设置实例的透明度。

⑤ 高级：用于同时设置实例的色调与透明度。

在最上面一层的第 20 帧按 F6 键插入关键帧，在“欢”字上单击鼠标右键，从快捷菜单中选择“任意变形”命令，文字四周出现 8 个黑色小方框，单击选中文字左边中间的那个小方点，然后向右拖动，使文字横向压缩直到翻转，如图 7.20 所示。

完成后在“欢”字上单击，在属性面板中将演样式设置为 Alpha，其值设置为 0%。

（5）创建运动渐变动画

把“欢”字移到场景右上角，选中第 1 帧，设置“补间”为“动画”，“缓动”为－100，如图 7.21 所示。

这样，这个字在 1～20 帧之间会由左下向右上方加速移动，在移动的过程中形状不断压缩直到翻转，颜色也逐渐变淡直至消失。

其他层的操作也是一样，因为每个文字起动的时间不一样，所以我们可以对每个字稍微做些调整。在“迎”层的第 3 帧和第 23 帧分别按 F6 键插入关键帧，在第 23 帧把“迎”字也做压缩翻转及设置 Alpha 值的处理，并拖到和“欢”字一样的位置，设置第 3 帧为运动渐变。这样“迎”字将比“欢”字晚 3 帧起动。其他文字用一样的方法，均比前一个字晚 3 帧起动，压缩

和颜色渐变等的设置方法一样。

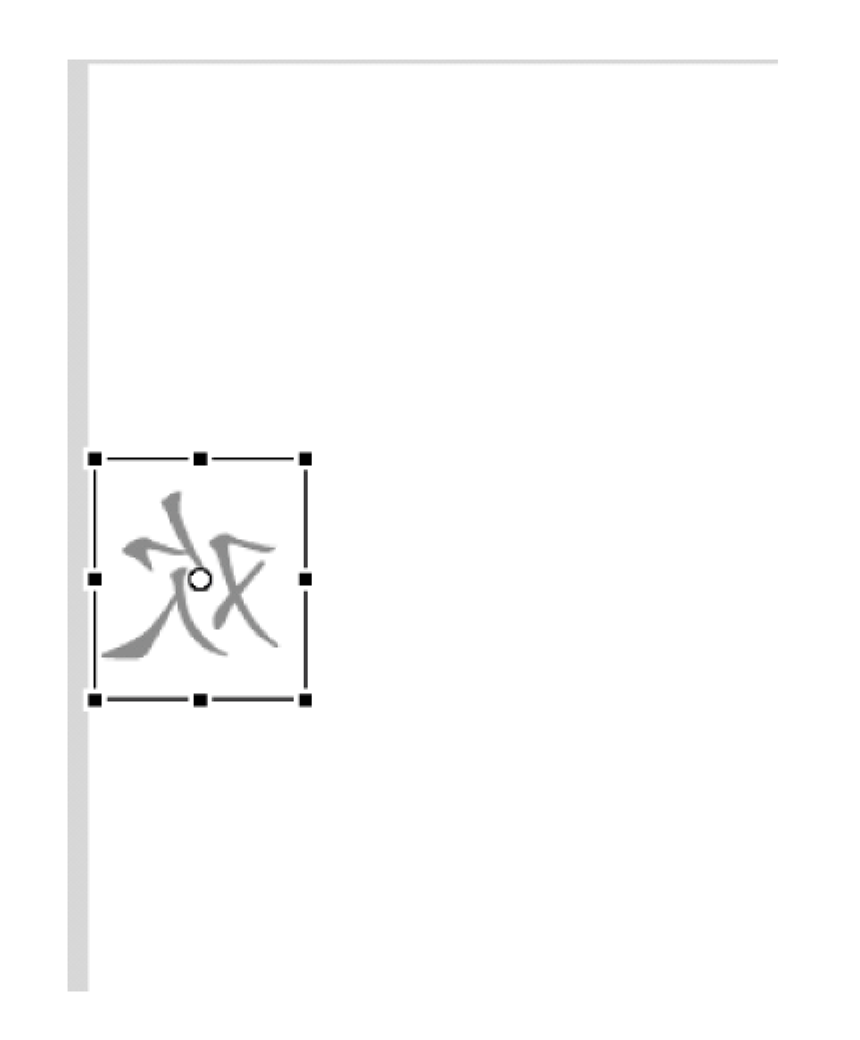



图 7.20　文字翻转



图 7.21　设置缓动

最终做好的时间轴窗口如图 7.22 所示。

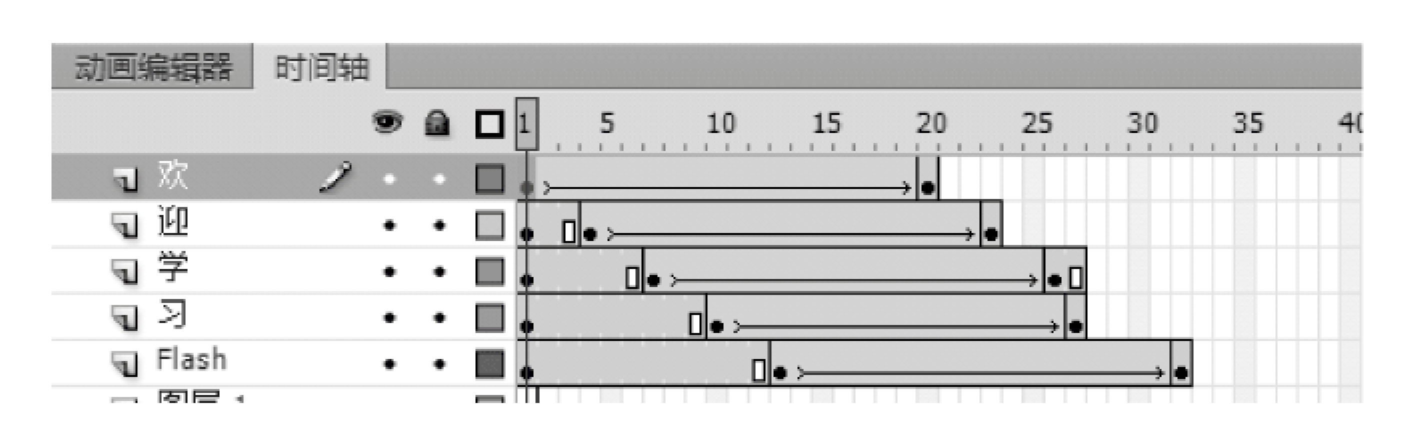


图 7.22　时间轴窗口

做好风吹动文字的动画，选择菜单栏中“控制”→“测试影片”命令来测试该动画效果，如图 7.23 所示。



图 7.23　动画效果

7.3.3　遮罩动画的制作

1. 什么是遮罩

遮罩动画是 Flash 的一种基本动画方式，制作遮罩动画至少需要两个图层，即遮罩层和被遮罩层。在时间轴上，位于上层的图层是遮罩层，这个遮罩层中的对象就像一个窗口一样，通过它的填充区域可以看到位于其下方的被遮罩层中的区域。而任何的非填充区域都是不透明的，被遮罩层在此区域中的图像将不可见。在一个遮罩动画中，“遮罩层”只有一个，“被遮罩层”可以有任意个。

2. 遮罩有什么用

遮罩主要有两种用途，一个作用是用在整个场景或一个特定区域，使场景外的对象或特定区域外的对象不可见；另一个作用是用来遮罩住某一元件的一部分，从而实现一些特殊的效果。

3. 创建遮罩的方法

（1）创建遮罩

在 Flash 中没有一个专门的按钮来创建遮罩，遮罩层其实是由普通图层转化过来的。只要在某个图层上单击鼠标右键，在弹出的菜单中把"遮罩"前打个钩，该图层就会生成遮罩层，"层图标"就会从普通层图标变为遮罩层图标，系统会自动把遮罩层下面的一层关联为"被遮罩层"，如果想关联更多层被遮罩，只要把这些层拖到被遮罩层下面就行了。

（2）构成遮罩和被遮罩层的元素

遮罩层中的图形对象在播放时是看不到的，遮罩层中的内容可以是按钮、影片剪辑、图形、位图、文字等，但不能使用线条，如果一定要用线条，可以将线条转化为"填充"。被遮罩层中的对象只能通过遮罩层中的对象被看到。在被遮罩层，可以使用按钮、影片剪辑、图形、位图、文字、线条等。

例 7.3　制作一个动画，产生文字在探照灯照射下逐步显示出来的效果。

第一步：新建一个动画文件，将第一个图层的名字改为"文字"

第二步：单击菜单"修改"→"文档"，设置背景色为黑色。在"文字"层的第 1 帧输入"欢迎光临"；在 15 帧处插入帧，将第一帧延伸至第 15 帧，如图 7.24 所示。



图 7.24　输入文字



图 7.25　新建元件

第三步：新建图形元件“圆”，如图 7.25 所示。

第四步：新建图层“灯”。将元件“圆”拖入第 1 帧的舞台左边，遮住左边的文字；在第 15 帧处插入关键帧，将“圆”实例移至舞台右边，遮住右边的文字，如图 7.26、图 7.27 所示。打开帧属性面板，在第 1 帧创建“传统补间动画”。

在图层“灯”上单击鼠标右键，选择“遮罩层”。

最后，用回车键或单击“控制”，选择“测试影片”，查看影片效果，如图 7.28 所示。

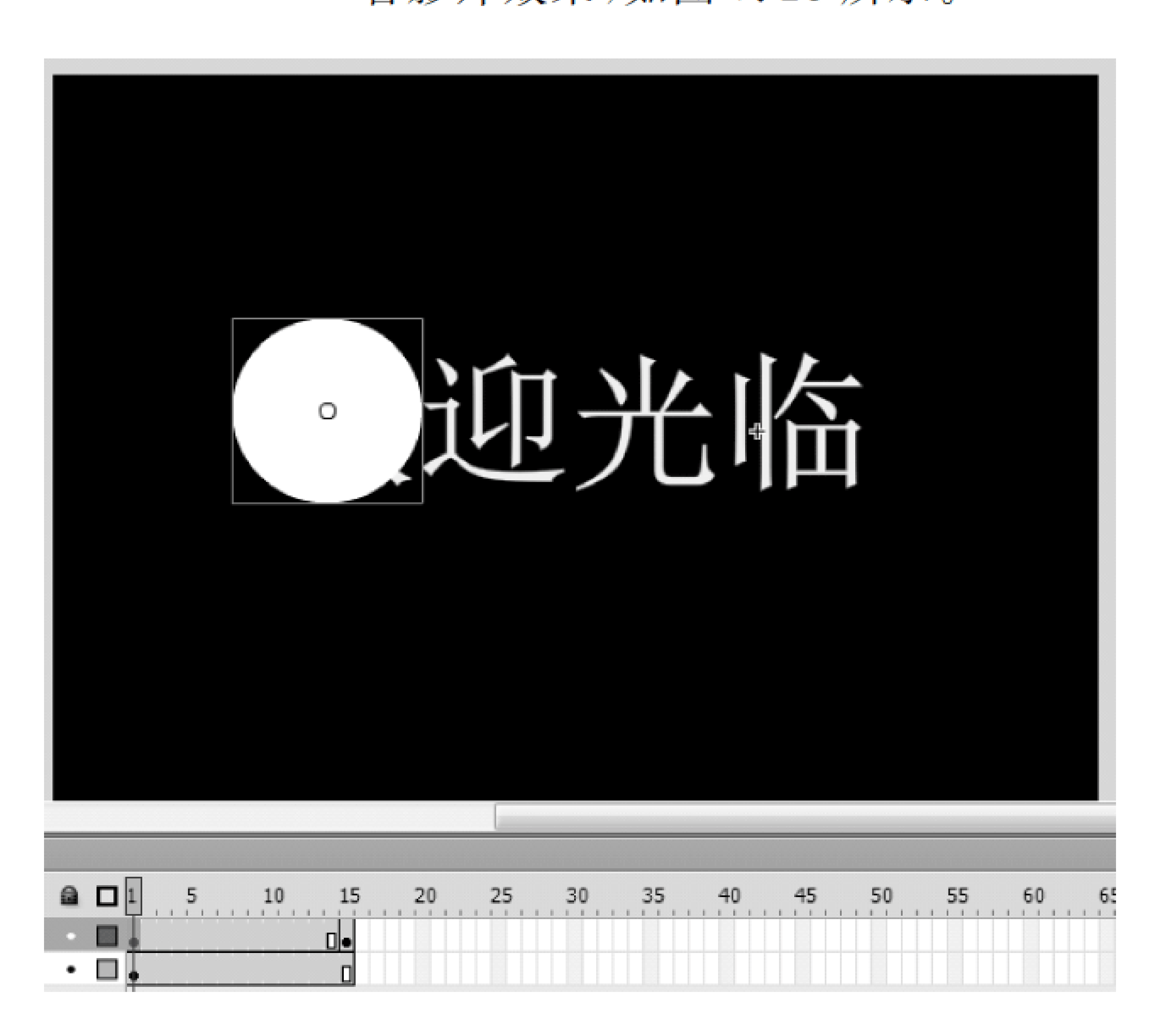


图 7.26　将“圆”拖入遮住左边字

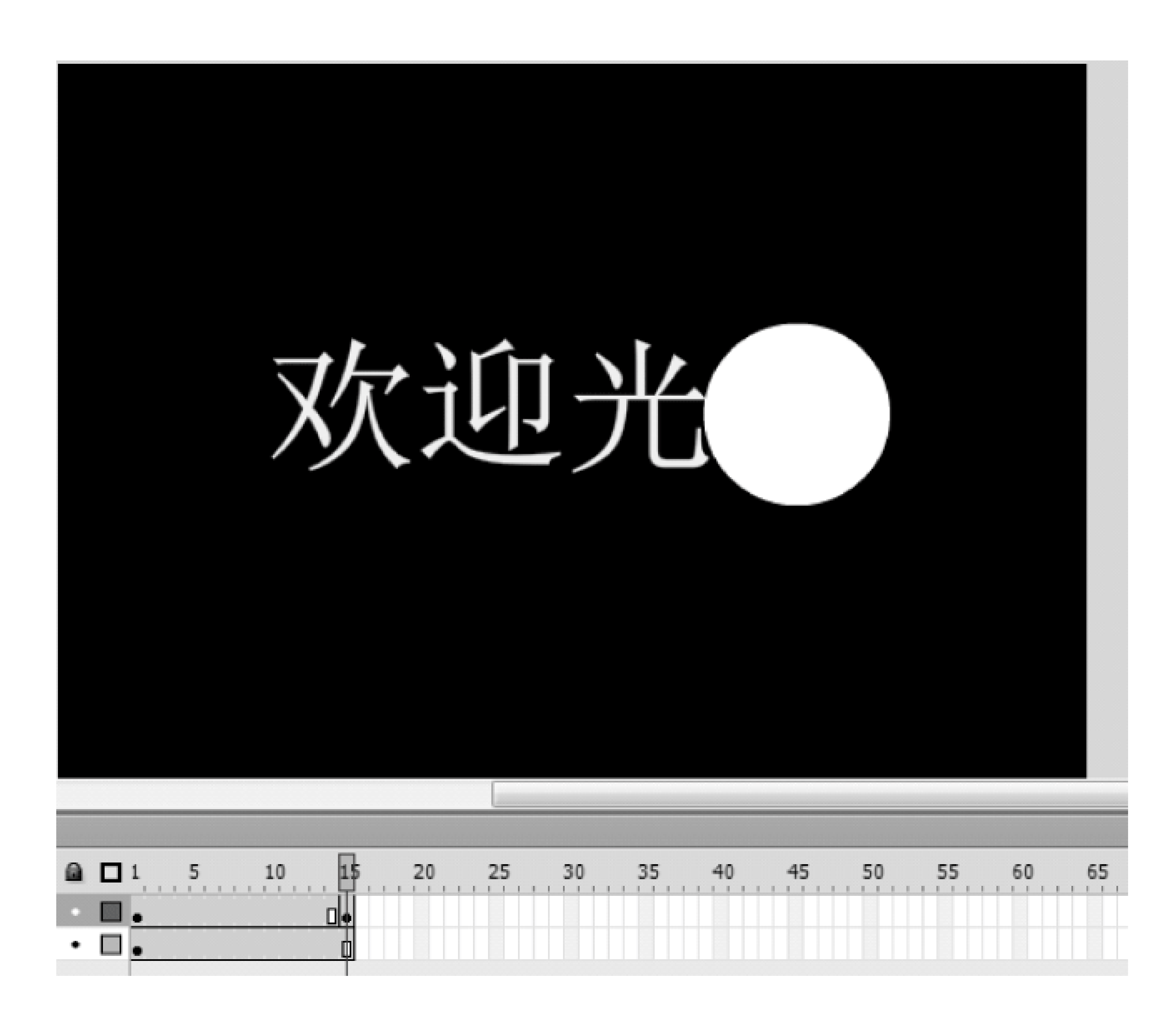


图 7.27　遮住右边字

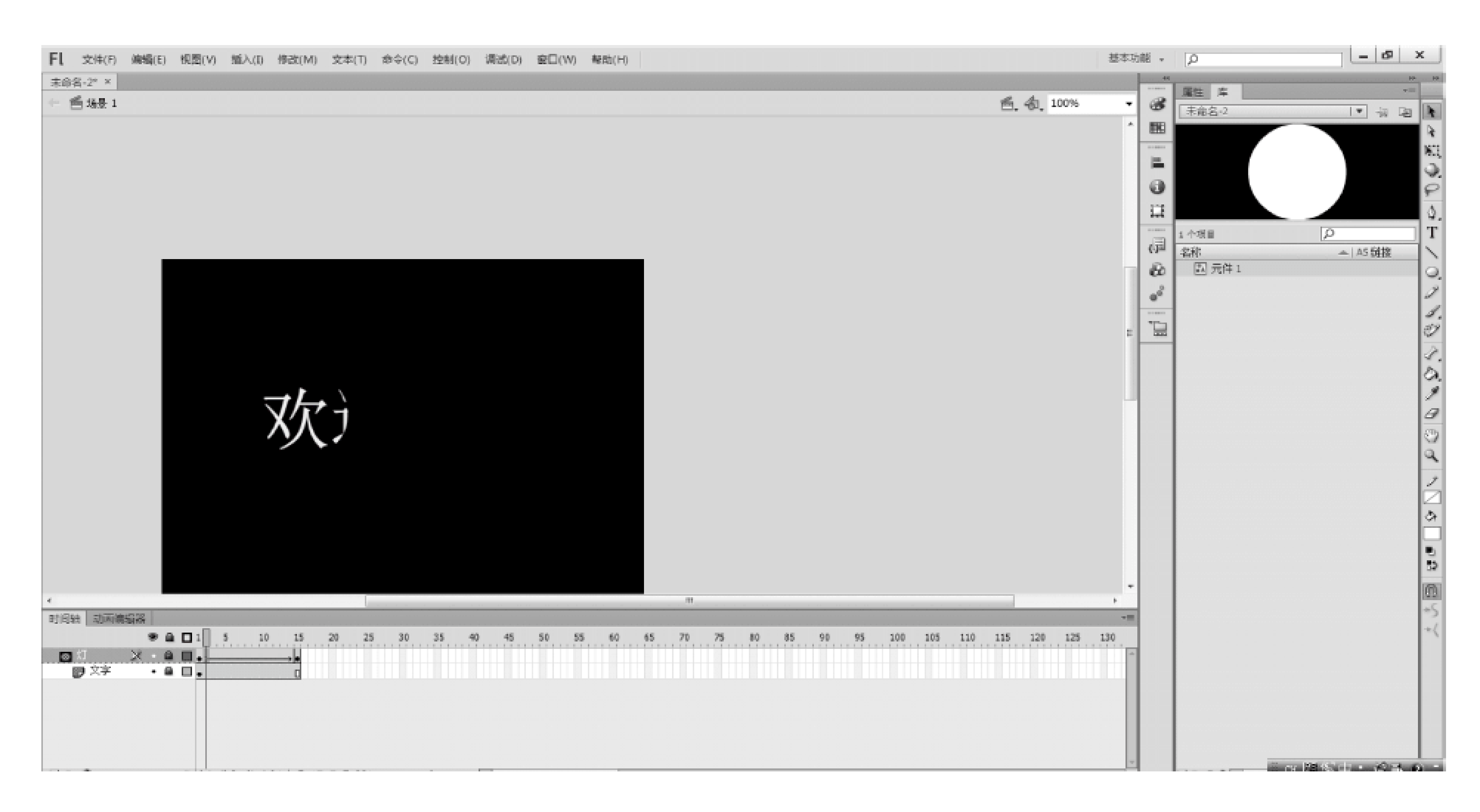


图 7.28　影片效果

7.3.4　引导层运动动画的制作

在一般情况下，运动动画产生补间动画时，都是按照最简单的方式产生的，即按直线从起点到终点方式产生，但事实上，很多运动并不按照直线轨迹运动，例如，雪花的飘落、小鸟的飞翔、鱼儿在大海里遨游等。当然，如果将曲线分解成一个个小的直线线段来完成也未尝不可，但在接点处的跳跃感仍会影响观看的效果。在 Flash 中能不能实现复杂的动画效果呢？答案是肯定的，这就是“引导路径动画”。将一个或多个层链接到一个运动引导层，使一个或多个对象沿同一条路径运动的动画形式被称为“引导路径动画”。在引导层中，利用可以产生曲线的工具绘制一条曲线，再利用属性面板的设置，使它作为运动对象或实体的运动轨迹。在演播时，这条曲线并不出现，但它的使用可以实现完美的曲线运动效果。

1. 创建引导路径动画的方法

(1) 创建引导层和被引导层

一个最基本“引导路径动画”由两个图层组成，上面一层是“引导层”，下面一层是“被引导层”。

(2) 引导层和被引导层中的对象

引导层是用来指示元件运行路径的，所以“引导层”中的内容可以是钢笔、铅笔、线条、椭圆工具、矩形工具或画笔工具等绘制出的线段。而“被引导层”中的对象是跟着引导线走的，可以使用影片剪辑、图形元件、按钮、文字等，但不能应用形状。

由于引导线是一种运动轨迹，不难想象，“被引导”层中最常见的动画形式是动作补间动画，当播放动画时，一个或数个元件将沿着运动路径移动。

(3) 向被引导层中添加元件

“引导动画”最基本的操作就是使一个运动动画“附着”在“引导线”上。所以操作时特别得注意“引导线”的两端，被引导的对象起始和终点的“中心点”一定要对准“引导线”的两个端头。

例 7.4　小球的移动。

（1）新建一个 Flash 空白文档

单击“文件”，选择“新建”，将宽高分别设为 400 和 300，背景为绿色，并将图层改名为小球，如图 7.29 所示。

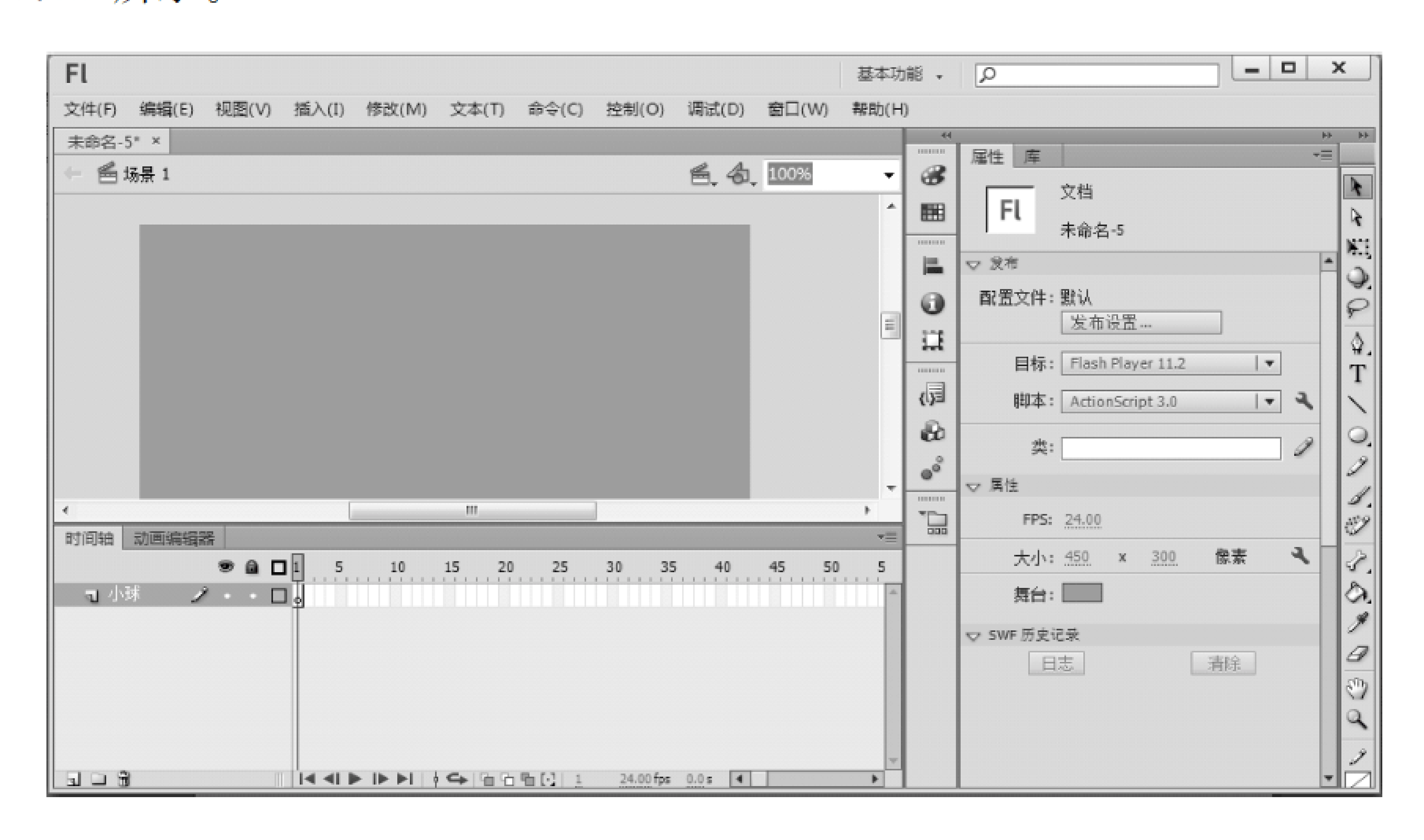


图 7.29　新建

（2）在小球图层中绘制一个小球

单击“椭圆工具”将笔触颜色改为无色，填充颜色为线形红色。然后在小球图层中绘制一个小球。注意：按下 Shift 键时，可画出标准的小球，如图 7.30 所示。

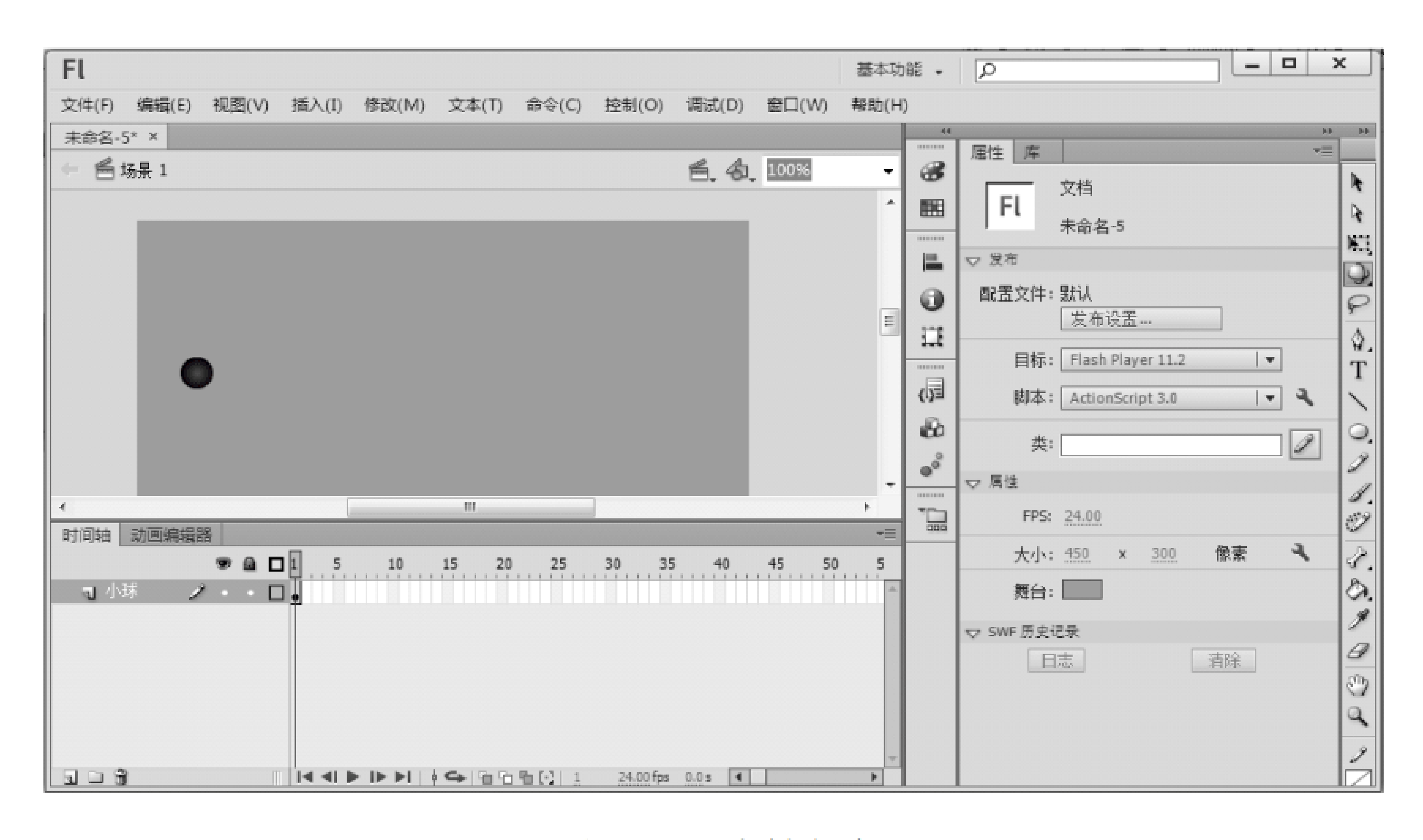


图 7.30　绘制小球

（3）制作小球的补间动画

在小球的第 30 帧插入一个关键帧，然后创建补间动画。

（4）添加运动引导层

右击小球图层，选择“添加传统运动引导层”，然后在引导层的第一帧，用线条工具绘制一个路径，如图 7.31 所示。

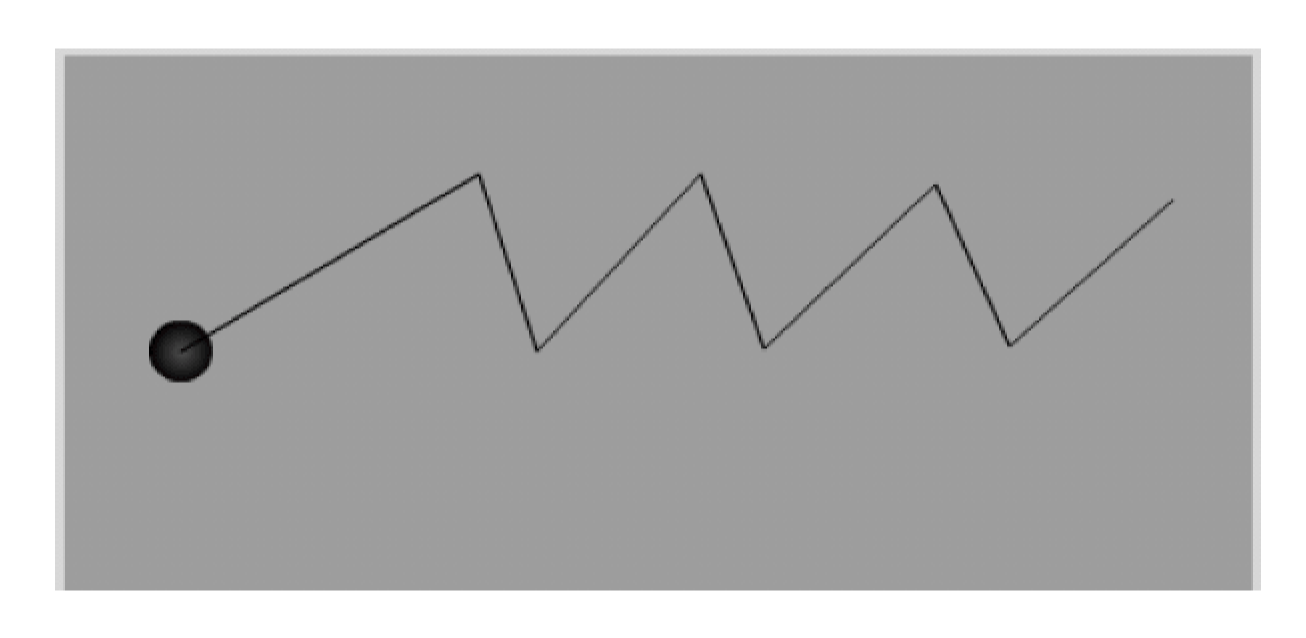


图 7.31　绘制路径

(5) 将小球移至路径的起点和终点

单击小球图层的第一帧，然后将小球移到路径的起点，如图 7.32 所示。

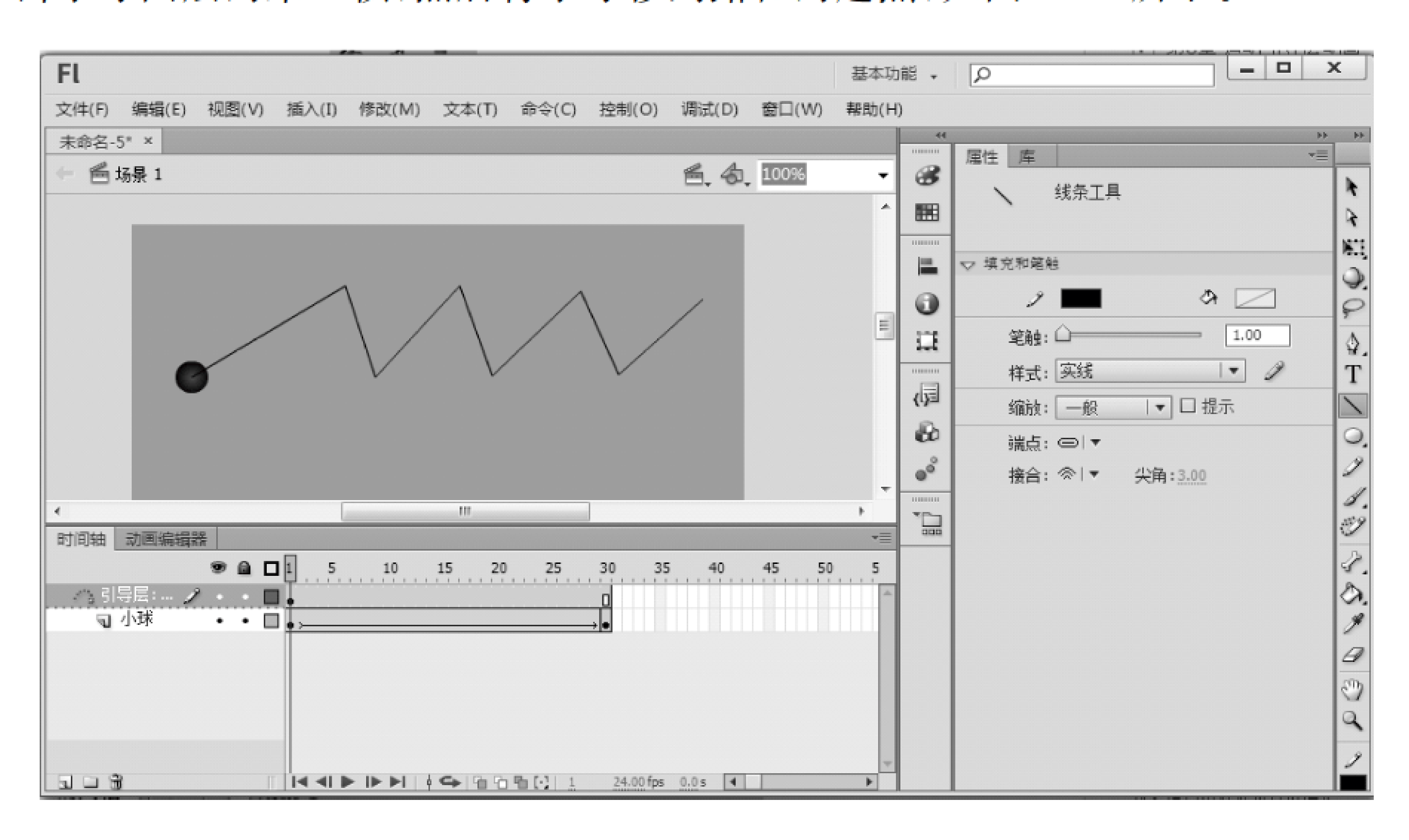


图 7.32　移动小球到起点

单击小球图层的最后一帧，然后将小球移动到路径的终点，如图 7.33 所示。

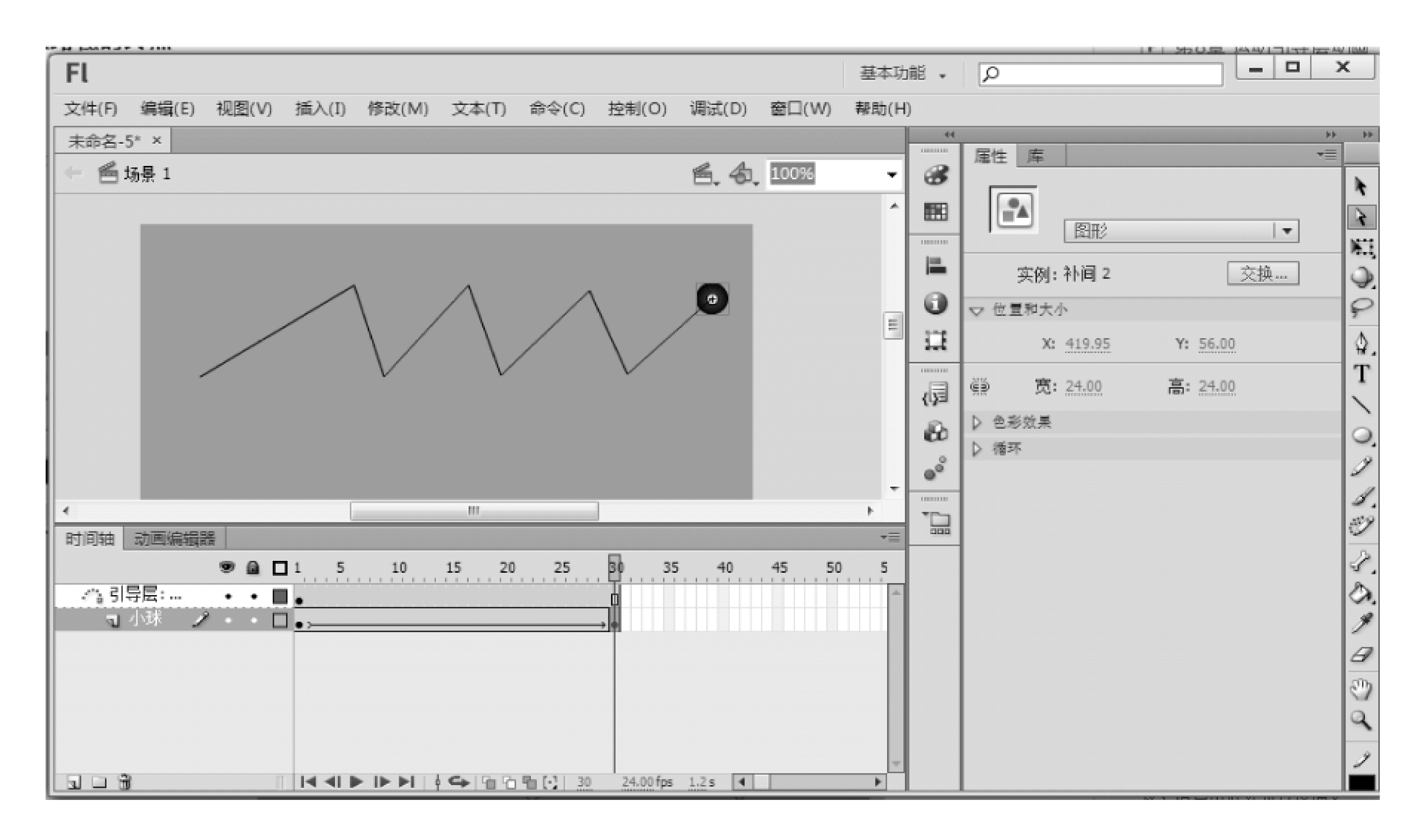


图 7.33　移动小球到终点

最后,同时按下 Ctrl+Enter 键看一下效果吧。

7.4 动画的测试、优化和发布

当所有的动画效果、交互效果、声音效果都被融入到一个 Flash 作品中以后,就要考虑如何将之输出为一个完整的文件了。在输出前,要对所有的动画效果、交互效果及声音效果做一个整体的测试,如果存在问题,就要进行调试、排除。

测试除了解决动画中存在的问题以外,还有一项重要的功能,就是优化。优化后的影片体积较小,可以达到最佳的传播效果。用 Flash 只做到动画是 fla 格式的,所以在动画制作完成后,需要将 fla 格式的文件发布成 swf 格式的文件(能被 Flash 播放器播放的动画文件),用于网页播放。

7.4.1 动画的测试

Flash CS 提供了强大的测试功能,因为 Flash 动画一般是在网格上播放,所以文件的大小对动画的播放流畅度影响很大,如果文件太大,浏览者很可能没有耐心等待动画下载完毕。作为优秀的动画设计师,不但要有全面的设计技术、敏锐的艺术感觉、新鲜的创意和创造力,还应掌握用最小的文件展示最完美的动画效果。

可以在两种环境下测试动画,一种为动画编辑环境,另一种为动画测试环境。下面就针对两种测试环境的特点,分别进行介绍。

1. 在动画编辑环境下测试影片

在动画编辑环境下,用户按 Enter 键可以对动画进行简单的测试,但动画中的影片剪辑元件、按钮元件等交互效果均不能得到测试。而且在动画编辑模式下,测试动画得到的动画速度比输出或优化后的动画速度慢。在动画编辑环境下通过设置,可以对按钮元件及简单的帧动作(play,stop,gotoplay 和 gotoandstop)进行测试。

(1) 按钮元件测试

要在动画编辑环境下测试按钮元件,选择“控制”→“启用简单按钮”命令。此时按钮将做出与最终动画中一样的响应,包括这个按钮所附加的脚本语言。

(2) 简单帧动作测试

要在动画编辑环境下测试简单的帧动作,选择“控制”→“启用简单帧动作”命令。

2. 在动画测试环境下测试

测试动画与场景主要通过“控制”菜单实现。执行“控制”→“测试影片”命令,Flash 将自动导出当前动画中的所有场景,然后将文件在新窗口中打开。

7.4.2 优化动画文件

随着动画文件大小的增大,它的下载和回放时间也会增加。作为发布过程中的一部分,Flash 自动对动画执行一些优化,例如,它可以在导出时检测重复的形状,在文件中只把它

们放一次,并且把嵌套组合转换成单个组合。

在导出动画之前,可以使用多种策略来减小文件的大小,从而对其进行进一步的优化。在动画发布的时候,也可以把它压缩成文件。

优化的主题策略包括以下4点。

1. 总体上优化

(1) 多次出现的元素,使用元件、动画或者其他对象。

(2) 只要可能,就要使用补间动画,补间动画与一系列的关键帧相比,占用的文件空间更少。

(3) 对于动画序列,要使用影片剪辑而不是图形元件。

(4) 限制每个关键帧中的改变的区域,在尽可能小的区域中执行动作。

(5) 避免使用动画位图元素,使用位图图像作为背景或者静态元素。

(6) 对于声音,尽可能使用MP3这种占用空间较小的声音格式,如非必要,不要添加太长的声音文件。

2. 优化元素和线条

(1) 尽量组合元素。

(2) 使用层把随动画过程改变的元素和不随动画过程改变的元素分开。

(3) 限制特殊线条类型的数量,如虚线、点状线、锯齿状线等,实线所需的内存空间较小,铅笔工具生成的线条比画笔笔触生成的线条所需的内存空间更小。

3. 优化文本和字体

(1) 限制字体的数量,尽量少使用嵌入字体,因为它们会增加文件的大小。

(2) 对于"嵌入字体"选项,只选中需要的字符,不包括所有字体。

4. 优化颜色

(1) 在元件属性检查器中,使用"颜色"菜单创建一个元件的具有不同颜色的多个实例。

(2) 使用混色器来使影片的颜色调色板与浏览器的调色板匹配。

(3) 尽量少使用渐变色,使用渐变色填充区域比使用纯色填充大概多需要50B空间。

(4) 尽量少使用Alpha透明度,它会减慢回放速度。

7.4.3 文件的发布

可以将Flash动画发布成多种格式,而在发布之前需进行必要的发布设置,定义发布的格式及相应的设置,达到最佳效果。在"发布设置"对话框中,可以一次性发布多种格式,且每种格式均保存为指定的发布设置,可以拥有不同的名字。接下来介绍发布动画的方法。

1. 发布设置

执行"文件"→"发布设置"命令。可以选择发布后输出的文件格式。默认的是Flash和HTML格式,因此在对话框中就有相应的选项卡。还可以选择发布成为GIF图像、JPEG

图像、PNG 图像、Windows 放映文件、Macintosh 放映文件。如果用户选中其他输出格式，自然也会在右边打开相应的选项卡。

2. 发布预览

执行“文件”→“发布预览”命令，该命令可以使发布的文件格式在默认打开的应用程序中打开预览，可以预览的文件格式类型是以“发布设置”对话框中的设置为基础的。

3. 发布动画

当制作完动画并对预览效果满意，就可以执行“文件”→“发布”命令。默认情况下，发布完后的文件以所选的文件类型的扩展名保存在和源文件同一目录下。

7.4.4 动画文件的导出

将 Flash 动画优化并测试后，就可以利用导出命令将动画导出为其他文件格式。除了可以保存.fla 格式的源文件，以及发布成为 Flash 播放文件.swf 格式或网页文件.html 格式外，还可以生成更多其他格式的文件。

在“文件”→“导出”菜单中包含三个导出命令：“导出图像”、“导出所选内容”和“导出影片”。

1. 导出图像

选择“导出图像”弹出“导出图像”对话框，可以选择导出文件类型为图像格式文件，可以保存为.dxf、.bmp、.jpg、.png 等格式文件。

2. 导出影片

选择“导出影片”弹出“导出硬片”对话框。可从“保存类型”下拉列表中选择。可以保存为.spl，.avi，.mov，.gif 等格式的文件。在“导出影片”对话框中保存的文件应注意以下选项的特点。

(1) Flash 影片(*.swf)：导出的文件是动态 swf 文件，只有在安装了 Flash 播放器的浏览器中才能播放，这也是 Flash 动画的默认保存文件类型。

(2) WAV 音频文件(*.wav)：仅导出影片中的声音文件。

(3) Adobe Illustrator 序列文件(*.ai)：保存影片中每一帧的矢量信息，在保存时可以选择编辑软件的版本，然后在 Adobe Illustrator 中进行编辑。

(4) GIF 动画(*.gif)：保存影片中每一帧的信息组成一个庞大的动态 GIF 动画，此时可以认为 Flash 是制作 GIF 动画的软件。

(5) JEPG 序列文件(*.jpg)：将影片中每一帧的图像一次导出为多个.jpg 文件。

小结

本章从 Flash CS6 的基础知识入手，介绍了 Flash CS6 的工作界面的组成及各模块的基本使用方法。讲解了 Flash 基本概念、基础动画设计及动画文件的测试、优化和导出。

习题

一、填空题

1. Flash CS6 源文件的扩展名是________。

2. 要直接在舞台上预览动画效果，应该按快捷键________。

3. 在 Flash CS6 程序窗口中，对动画内容进行编辑的整个区域称为________，用户可以在整个区域内对对象进行编辑绘制。________用于显示动画文件的内容，供用户对对象进行浏览、绘制和编辑，默认情况下它为白色。

4. 元件有________、________和________ 3 种类型。

5. 在 Flash 中要创建元件，一般有 2 种方法，一种方法是________，另一种方法是首先创建一个________，然后________。

二、简答题

1. 什么是逐帧动画？逐帧动画有什么特点？

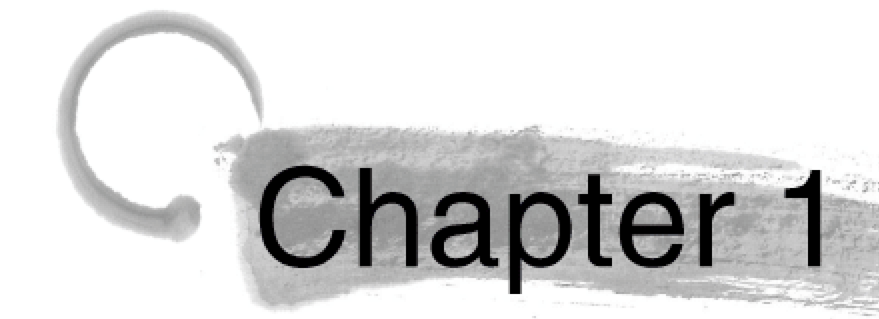
2. 什么是遮罩动画？遮罩动画有什么特点？

3. 什么是引导层动画？引导层动画有什么特点？

三、上机实验

请在互联网上搜集素材，做出一个雪花纷飞的场景。

第 二 部 分



Chapter 1 HTML & CSS Reference

1.1 HTML Reference

Ordered Alphabetically

Tag	**Description**
<!--...-->	Defines a comment
<!DOCTYPE>	Defines the document type
<a>	Defines an anchor
<abbr>	Defines an abbreviation
<acronym>	Defines an acronym
<address>	Defines contact information for the author/owner of a document
<applet>	Deprecated. Defines an embedded applet
<area />	Defines an area inside an image-map
<b>	Defines bold text
<base />	Defines a default address or a default target for all links on a page
<basefont />	Deprecated. Defines a default font, color, or size for the text in a page
<bdo>	Defines the text direction
<big>	Defines big text
<blockquote>	Defines a long quotation
<body>	Defines the document's body
 	Defines a single line break
<button>	Defines a push button
<caption>	Defines a table caption
<center>	Deprecated. Defines centered text
<cite>	Defines a citation
<code>	Defines computer code text
<col />	Defines attribute values for one or more columns in a table
<colgroup>	Defines a group of columns in a table for formatting
<dd>	Defines a description of a term in a definition list
<del>	Defines deleted text
<dir>	Deprecated. Defines a directory list

<div>	Defines a section in a document
<dfn>	Defines a definition term
<dl>	Defines a definition list
<dt>	Defines a term (an item) in a definition list
<em>	Defines emphasized text
<fieldset>	Defines a border around elements in a form
<font>	Deprecated. Defines font, color, and size for text
<form>	Defines an HTML form for user input
<frame />	Defines a window (a frame) in a frameset
<frameset>	Defines a set of frames
<h1> to <h6>	Defines HTML headings
<head>	Defines information about the document
<hr />	Defines a horizontal line
<html>	Defines an HTML document
<i>	Defines italic text
<iframe>	Defines an inline frame
<img />	Defines an image
<input />	Defines an input control
<ins>	Defines inserted text
<isindex>	Deprecated. Defines a searchable index related to a document
<kbd>	Defines keyboard text
<label>	Defines a label for an input element
<legend>	Defines a caption for a fieldset element
	Defines a list item
<link />	Defines the relationship between a document and an external resource
<map>	Defines an image-map
<menu>	Deprecated. Defines a menu list
<meta />	Defines metadata about an HTML document
<noframes>	Defines an alternate content for users that do not support frames
<noscript>	Defines an alternate content for users that do not support client-side
scripts	**STF**
<object>	Defines an embedded object
	Defines an ordered list
<optgroup>	Defines a group of related options in a select list
<option>	Defines an option in a select list
<p>	Defines a paragraph
<param />	Defines a parameter for an object
<pre>	Defines preformatted text
<q>	Defines a short quotation
<s>	Deprecated. Defines strikethrough text
<samp>	Defines sample computer code
<script>	Defines a client-side script
<select>	Defines a select list (drop-down list)

<small>	Defines small text
<span>	Defines a section in a document
<strike>	Deprecated. Defines strikethrough text
<strong>	Defines strong text
<style>	Defines style information for a document
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<table>	Defines a table
<tbody>	Groups the body content in a table
<td>	Defines a cell in a table
<textarea>	Defines a multi-line text input control
<tfoot>	Groups the footer content in a table
<th>	Defines a header cell in a table
<thead>	Groups the header content in a table
<title>	Defines the title of a document
<tr>	Defines a row in a table
<tt>	Defines teletype text
<u>	Deprecated. Defines underlined text
	Defines an unordered list
<var>	Defines a variable part of a text
<xmp>	Deprecated. Defines preformatted text

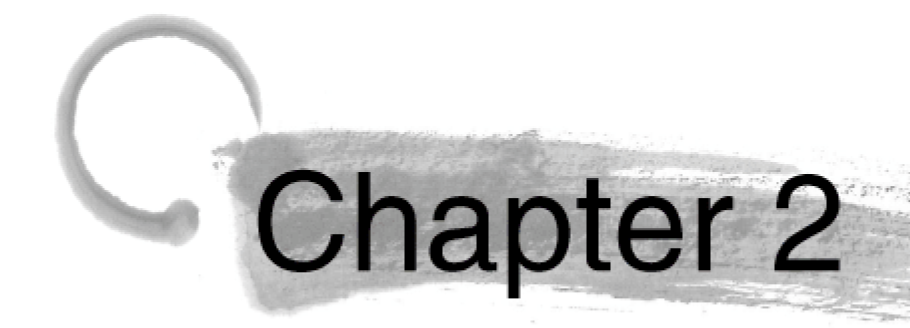
1.2 CSS Reference

Ordered Alphabetically

Property	**Description**
background	Sets all the background properties in one declaration
background-attachment	Sets whether a background image is fixed or scrolls with the rest of the page
background-color	Sets the background color of an element
background-image	Sets the background image for an element
background-position	Sets the starting position of a background image
background-repeat	Sets how a background image will be repeated
border	Sets all the border properties in one declaration
border-bottom	Sets all the bottom border properties in one declaration
border-bottom-color	Sets the color of the bottom border
border-bottom-style	Sets the style of the bottom border
border-bottom-width	Sets the width of the bottom border
border-color	Sets the color of the four borders
border-collapse	Specifies whether or not table borders should be collapsed
border-left	Sets all the left border properties in one declaration

border-left-color	Sets the color of the left border
border-left-style	Sets the style of the left border
border-left-width	Sets the width of the left border
border-right	Sets all the right border properties in one declaration
border-right-color	Sets the color of the right border
border-right-style	Sets the style of the right border
border-right-width	Sets the width of the right border
border-spacing	Specifies the distance between the borders of adjacent cells
border-style	Sets the style of the four borders
border-top	Sets all the top border properties in one declaration
border-top-color	Sets the color of the top border
border-top-style	Sets the style of the top border
border-top-width	Sets the width of the top border
border-width	Sets the width of the four borders
bottom	Sets the bottom margin edge for a positioned box
caption-side	Specifies the placement of a table caption
clear	Specifies which sides of an element where other floating elements are not allowed
clip	Clips an absolutely positioned element
color	Sets the color of text
content	Used with the：before and：after pseudo-elements， to insert generated content
counter-increment	Increments one or more counters
counter-reset	Creates or resets one or more counters
cursor	Specifies the type of cursor to be displayed
direction	Specifies the text direction/writing direction
display	Specifies the type of box an element should generate
empty-cells	Specifies whether or not to display borders and background on empty cells in a table
float	Specifies whether or not a box should float
font	Sets all the font properties in one declaration
font-family	Specifies the font family for text
font-size	Specifies the font size of text
font-style	Specifies the font style for text
font-variant	Specifies whether or not a text should be displayed in a small-caps font
font-weight	Specifies the weight of a font
height	Sets the height of an element
left	Sets the left margin edge for a positioned box
letter-spacing	Increase or decrease the space between characters in a text
line-height	Sets the line height
list-style	Sets all the properties for a list in one declaration
list-style-image	Specifies an image as the list-item marker
list-style-position	Specifies where to place the list-item marker
list-style-type	Specifies the type of list-item marker

margin	Sets all the margin properties in one declaration
margin-bottom	Sets the bottom margin of an element
margin-left	Sets the left margin of an element
margin-right	Sets the right margin of an element
margin-top	Sets the top margin of an element
max-height	Sets the maximum height of an element
max-width	Sets the maximum width of an element
min-height	Sets the minimum height of an element
min-width	Sets the minimum width of an element
orphans	Sets the minimum number of lines that must be left at the bottom of a page when a page break occurs inside an element
outline	Sets all the outline properties in one declaration
outline-color	Sets the color of an outline
outline-style	Sets the style of an outline
outline-width	Sets the width of an outline
overflow	Specifies what happens if content overflows an element's box
padding	Sets all the padding properties in one declaration
padding-bottom	Sets the bottom padding of an element
padding-left	Sets the left padding of an element
padding-right	Sets the right padding of an element
padding-top	Sets the top padding of an element
page-break-after	Sets the page-breaking behavior after an element
page-break-before	Sets the page-breaking behavior before an element
page-break-inside	Sets the page-breaking behavior inside an element
position	Specifies the type of positioning for an element
quotes	Sets the type of quotation marks for embedded quotations
right	Sets the right margin edge for a positioned box
table-layout	Sets the layout algorithm to be used for a table
text-align	Specifies the horizontal alignment of text
text-decoration	Specifies the decoration added to text
text-indent	Specifies the indentation of the first line in a text-block
text-shadow	Specifies the shadow effect added to text
text-transform	Controls the capitalization of text
top	Sets the top margin edge for a positioned box
unicode-bidi	sets the direction of text
vertical-align	Sets the vertical alignment of an element
visibility	Specifies whether or not an element is visible
white-space	Specifies how white-space inside an element is handled
widows	Sets the minimum number of lines that must be left at the top of a page when a page break occurs inside an element
width	Sets the width of an element
word-spacing	Increases or decreases the space between words in a text
z-index	Sets the stack order of an element



HTML

2.1 Getting to Know HTML

2.1.1 Hello, World－creating Your First HTML File

To create HTML files in Windows OS we are going to use Notepad—it ships with every copy of Windows. If you have got your own favorite editor that runs on Windows, that's fine too; just make sure you can create a plain text file with an ".html" extension.

Create your first HTML file step by step (take Windows 7 as example).

Step 1:

Open the Start menu and navigate to Notepad.

You'll find the Notepad application in Accessories. The easiest way to get there is to click on the "Start" menu, then on "All Programs", then "Accessories". You'll see Notepad listed there.

Step 2:

Open Notepad.

Once you've located Notepad in the Accessories folder, go ahead and click on it. You'll see a blank window ready for you to start typing HTML.

Step 3 (optional but recommended):

Don't hide extensions of well known file types.

By default Windows 7's File Explorer hides the file extensions of well known file types. For example, a file named "hello.html" will be shown in the Explorer as "hello" without its ".html" extension.

It's much less confusing if Win7 shows you these extensions, so let's change your folder options so you can see the file extensions.

First, in any Explorer window select "Folder Options…" from the Tools menu.

Next, in the "View" tab, under "Advanced settings", scroll down until you see "Hide extensions for known file types" and uncheck this option.

That's it. Click on the OK button to save the preference and you'll now see the file extensions in the Explorer.

Now, type the following code in the blank window of Notepad:

Example 2.1

Then, save your work by choosing "Save" from the File menu and you'll see a "Save As" dialog box. First, click Desktop icon on left column of dialog box, then choose "All Files" in "Save as type" drop-down box (otherwise Notepad will add a ".txt" to your filename). Next, enter "index.html" as the file name and click on the Save button.

Open your Web page in a browser.

Double click the index.html file you created (located in desktop), success! The browser has shown the web page for you as the Figure 2.1.

Congratulations, you've just written your first HTML!

说明：为避免代码及图片重复，本章所提到的所有例子及图片，均对应中文第 2 章 HTML 相同的序号及图片。

2.1.2 Terms of HTML

You may notice the <html>, </html>, <h1>, <body> in Hello, World HTML document, they are used to mark up the text in HTML. Now, let's take a close look at how HTML tags work in Example 2.2.

Example 2.2.

In Example 2.2, <h1> and </h1> are called tags with <h1> is opening tag and </h1> is closing tag. We usually put some pieces of content within opening tag and closing tag. Here, the content is "Hello, World". We use tags to tell the browser that our content, "Hello, World", is a top level heading (that is, heading level one). The whole shebang of Example 2.2 is called an element. In this case we can call it the <h1> element. An element consists of the enclosing tags and the content in between.

To tell the browser about the structure of your web page, use pairs of tags around your content. Remember the following formula:

Element = Opening Tag + Content + Closing Tag

Tags consist of the tag name surrounded by angle brackets, that is, the < and > character. As opening tag, <h1> begins the element, and closing tag </h1> ends the element. We know </h1> is a closing tag because it comes after the content, and it's got a "/" before the "h1". All closing tags have a "/" in them.

tag 标签，标记

opening tag or start tag 起始标签

closing tag or end tag 终止标签

element 元素

2.1.3 HTML and HTML5

The original language of the World Wide Web is HTML (HyperText Markup

Language), often referred to by its current version, HTML 4.01 or just HTML4 for short. HTML was originally an application of SGML (Standard Generalized Markup Language), a sort of meta-language for making markup languages. SGML is quite complicated, and in practice most browsers do not actually follow all of its oddities. HTML as actually used on the web is best described as a custom language influenced by SGML.

Another important thing to note about HTML is that all HTML user agents (this is a term for programs that read HTML, including web browsers, search engine web crawlers, and so forth) have extremely tolerant error handling. Many technically illegal constructs, like misnested tags or bad attribute names, are allowed or safely ignored. This error-handling is relatively consistent between browsers. But there are lots of differences in edge cases, because this error handling behavior is not documented or part of any standard. This is why it is a good idea to validate your documents.

HTML5 is the latest standard for HTML.

HTML5 was designed to replace both HTML 4, XHTML.

Its core aims have been to improve the language with support for the latest multimedia while keeping it easily readable by humans and consistently understood by computers and devices.

HTML5 is also cross-platform. It is designed to work whether you are using a PC, or a Tablet, a Smartphone, or a Smart TV.

HTML 超文本置标语言

meta 元,…的…

agent 代理程序

parser 解析器,分析器

2.1.4 Get Ready for HTML

Many pages on the Internet contain "bad" HTML.

The following HTML code will work just fine if you view it in a browser (even if it does not follow the HTML rules):

Example 2.3

To standardize the HTML codes, the following 4 major rules must be abided by:

1. HTML Elements Must Be Properly Nested

In HTML, some elements can be improperly nested within each other, like this:

Example 2.4

In HTML, all elements must be properly nested within each other, like this:

Example 2.5

Note: A common mistake with nested lists, is to forget that the inside list must be within <li> and tags.

This is wrong:

Example 2.6

This is correct:

Example 2.7

Notice that we have inserted a tag after the tag in the "correct" code example.

2. HTML Elements Must Always Be Closed

Non-empty elements must have a closing tag.

This is wrong:

Example 2.8

This is correct:

Example 2.9

Empty elements must also be closed.

This is wrong:

Example 2.10

This is correct:

Example 2.11

3. HTML Elements Must Be In Lower Case

Tag names and attributes must be in lower case.

This is wrong:

Example 2.12

This is correct:

Example 2.13

4. HTML Documents Must Have One Root Element

All HTML elements must be nested within the <html> root element. Child elements must be in pairs and correctly nested within their parent element.

The basic document structure is:

Example 2.14

The HTML element is the outermost element in HTML and HTML documents. The HTML element is also known as the root element.

The <html> tag tells the browser that this is an HTML document.

The head element is a container for all the elements within the head section. Elements inside <head> can include scripts, instruct the browser where to find style sheets, provide meta information, and more. The following tags can be added to the head section: <base>, <link>, <meta>, <script>, <style>, and <title>. The <title> tag defines the title of the document, and is the only required element in the head section.

The body element contains all the contents displayed in web browser, such as text, hyperlinks, images, tables, lists, etc.

A simple HTML document, with the minimum of required tags:

Example 2.15

Web page of Example 2.15 is shown as Figure 2.2

Some more HTML syntax rules will be instructed in related sections.

2.2 Text Elements

From this section, we begin to learn basic HTML tags and attributes.

2.2.1 Headings

Here is Example 2.16 to show you how headings' tag work:

Example 2.16

And Figure 2.3 is how Example 2.16 looks in the browser:

The <h1> to <h6> tags are used to define HTML headings.

Syntax of Headings:

```
<h# align="left|center|right">headings</h#>
```

(from 1 to 6) is the size of headings, <h1> defines the largest heading and <h6> defines the smallest heading.

"align" is an optional attribute, which specifies the alignment of a heading. "|" in syntax means "or", i.e. multiple-choice.

Attributes:

Tags can also have attributes, which are extra bits of information. Attributes appear inside the opening tag and their value is always inside quotation marks and must be in lower case. They look something like <tag attribute="value">content…</tag>. Tags become more useful once you add attributes.

Example 2.17

The browser shows Example 2.17 as Figure 2.4.

In comparison with Example 2.2, let's take a look at the element's structure once more in Example 2.18.

Example 2.18

<h1> is opening tag and </h1> is closing tag, and "Heading 1 is on the left." is content in between. Now we can add optional attributes in opening tag to enrich the element.

Note 1: Tag <h#> and tag <head> are different.

Note 2: The "align" attribute of the <h1> to <h6> elements was deprecated in HTML 4.01, and is not supported in HTML 1.0 Strict DTD. Use CSS instead.

2.2.2 Title

The <title> tag defines the title of the document.

The title element:

- defines a title in browser's title bar.
- provides a title for the page when it is added to favorites.
- displays a title for the page in search-engine results.

Every HTML document must have a title element in the head section. Note that <title> element is required within <head> section rather than <body> section.

Authors should use the title element to identify the contents of a document. Since users often consult documents out of context, authors should provide context-rich titles. Thus, instead of a title such as "Introduction", which doesn't provide much contextual background, authors should supply a title such as "Introduction to Nordic Walking" instead.

Syntax of Title:

```
<title>title's text</title>
```

Replace the Example 2.17 <title> element content with "align attribute of <h> tag", as Example 2.19.

Example 2.19

The browser shows Example 2.19 as Figure 2.5.

2.2.3 Paragraph

The p element automatically creates some space before and after itself. The space is automatically applied by the browser, or you can specify it in a style sheet.

Syntax of paragraph:

```
<p align="left|center|right">text</p>
```

Example 2.20

The browser shows Example 2.20 as Figure 2.6.

Nothing special? Try to copy all the contents in browser's window of Figure 2.6, and then paste them to notepad.

Anything special? You may know why so many extra single line between paragraphs when you copy the contents from web page and then paste them to your notepad or Word application.

Some other default behaviors of paragraphs.

Example 2.21

The browser shows Example 2.21 as Figure 2.7.

From now on, this textbook will lead readers to construct a website for a fictional

club, Walker Club, step by step from what has been taught.

Project 2.1: Home page of Walker Club (index.html version 1).

The browser shows Project 2.1 as Figure 2.8.

2.2.4 Single Line Break

The
 tag inserts a single line break.

The
 element is an element that doesn't have any content. Why? Because it's just meant to be a linebreak, nothing else.
 isn't the only element like this; there are others, and we have a name for them: empty elements. In HTML the
 tag has no end tag. In HTML the
 tag must be properly closed, like this:
.

From p tag, we know that browsers ignore lines and spaces created by Enter key and spacebar.
 tag is used to force a line break.

Syntax of linebreak:

```
Text <br />
```

Insert Example 2.22 to <body> element, we can understand the function of
 tag.

Example 2.22

In respect of linebreak itself,

<p>text</p>

is equivalent to

text

got it?

In practice, use the
 tag to insert line breaks, not to create paragraphs.

Example 2.23

The browser shows Example 2.23 as Figure 2.9.

The <nobr> tag is used to disallow any line breaks in the text between the tags.

2.2.5 Horizontal Rule

The <hr> tag creates a horizontal line in an HTML page.

The hr element is an empty element and can be used to separate content in an HTML page.

Syntax of horizontal rule:

```
<hr align = "left|center|right" size = "pixels" width = "pixels or %" color = "color or color code" />
```

size: height of horizontal rule.

width: width of horizontal rule.

color: color of horizontal rule, use color or color code. Refer to Table 2.1 for some

common colors and color code.

2.2.6 Comment

The comment tag (<! --…-->) is used to insert a comment in the source code. A comment will be ignored by the browser. You can use comments to explain your code, which can help you when you edit the source code at a later date.

You can also store program-specific information inside comments. In this case they will not be visible for the user, but they are still available to the program. A good practice is to comment the text inside scripts and style elements to prevent older browsers, that do not support scripting or styles, from showing it as plain text.

Syntax of comment:

```
<! -- comments -- >
```

Example 2.24

The browser shows Example 2.24 as Figure 2.10.

2.2.7 Division

The <div> tag defines a division or a section in an HTML document.

The <div> tag is often used to group block-elements to format them with styles.

Syntax of division:

```
<div> content (text, image or table)</div>
```

Example 2.25

Project 2.2 is shown in the browser as Figure 2.11.

The div element is very often used with CSS to layout a web page.

2.3 Hyperlink

In web terms, a hyperlink is a reference (an address) to a resource on the web. Hyperlinks can point to any resource on the web: an HTML page, an image, a sound file, a movie, etc. An anchor is a term used to define a hyperlink destination inside a document. The HTML anchor element <a>, is used to define both hyperlinks and anchors.

The <a> element is usually referred to as a link or a hyperlink.

2.3.1 <a> Tag

Example 2.26 defines a link to Example 2.25 web page by using <a> tag.

Example 2.26

The browser shows Example 2.26 as Figure 2.12.

Syntax of anchor:

```
<a href = "url">Link text</a>
```

The most important attribute of the <a> element is the href attribute, which defines the link "destination address". The start tag contains attributes about the link. The element content (Link text) defines the part to be displayed.

Note: The element content doesn't have to be text. You can link from an image or any other HTML element.

By default, links will appear as follows in all browsers:

An unvisited link is underlined and blue. A visited link is underlined and purple. An active link is underlined and red.

2.3.2 Path and Directory

A path, the general form of a filename or of a directory name, specifies a unique location in a file system. A path points to a file system location by following the directory tree hierarchy expressed in a string of characters in which path components, separated by a delimiting character, represent each directory. The delimiting character is most commonly the slash ("/"), the backslash character ("\"), though some operating systems may use a different delimiter. Paths are used extensively in computer science to represent the directory/file relationships common in modern operating systems, and are essential in the construction of Uniform Resource Locators (URLs).

Systems can use either absolute or relative paths. A full path or absolute path is a path that points to the same location on one file system regardless of the working directory or combined paths. It is usually written in reference to a root directory. A relative path is a path relative to the working directory of the user or application, so the full absolute path may not need to be given.

Example 2.27 absolute path.

Example 2.28 relative path.

As can be seen from the above example, the absolute path contains the full path instead of a few directories contained within the absolute path.

The absolute path is typically used with the domain to point to Web elements that are on another domain. For example, link to the Google site—what need to do is to include the domain in the URL as Example 2.29.

Example 2.29

If referring to a Web element that is on the same domain, relative path is a good choice. Relative paths change depending upon what page the links are located on. There are 4 rules to creating a link using the relative path:

1. link to the file in the current directory

```
<a href = "filename.html"> Link text </a>
```

2. link to the file in the subdirectory

```
<a href = "subdirectory/filename.html"> Link text </a>
```

3. link to the file in the parent directory

```
<a href = "../filename.html"> Link text </a>
"../" means return to the parent directory
```

4. link to the file in the sibling directory

```
<a href = "../sibling directory/filename.html"> Link text </a>
"../sibling directory/" means return to the parent directory first, and then get in to the sibling directory
```

2.3.3 Organizing Website Directory Structure

Before start creating more HTML pages, it's time to get things organized. So far, there is only one file "index. html" in Project directory/folder. It will be much more manageable if the Web pages, graphics, and other resources are organized into a set of directories/folders. Now, the Walker Club is given a meaningful directory structure as Figure 2. 13 and Figure 2. 14.

From the 4 rules studied in previous section, the following 4 examples instruct how to create hyperlink with reference to Figure 2. 14.

Link from "article1. html" to "article2. html"

Example 2. 30

Link from "index. html" to "article2. html"

Example 2. 31

Link from "activity1. html" to "index. html"

Example 2. 32

Link from "article1. html" to "activity1. html"

Example 2. 33

2.3.4 The Target Attribute

The target attribute of <a> tag defines where the linked document will be opened. The code below (modified version of Example 2. 26) will open the document in a new

browser window:

Example 2.34

Compare Example 2.26 with its modified version Example 2.34. When "Link to Example 2.25" is clicked in Example 2.26 web page, the browser shows Example 2.25 web page in current window (it replaces the Example 2.26 web page). However, when "Link to Example 2.25" is clicked in Example 2.34 web page, Example 2.25 web page is opened in a new browser window (Example 2.34 web page is still there). The benefit is obvious-visitors open a link in a new browser window, so that they do not have to leave your Web site.

2.3.5 Link to A Location on the Same Page

The term link or hyperlink is used when the <a> element points to a resource. The term anchor is used when the <a> elements define an address inside a document. The hyperlink has been discussed in previous section, and the anchor will be discussed in this section.

When the name attribute is used, the <a> element defines a named anchor inside an HTML document.

Named anchor syntax:

```
<a name = "label"> Any content </a>
```

The link syntax to a named anchor:

```
<a href = "#label"> Link text </a>
```

The # in the href attribute defines a link to a named anchor.

Example 2.35

The browser shows Example 2.35 as Figure 2.15.

Named anchor are not displayed in any special way. They are invisible to the reader. Named anchors are often used to create "table of contents" at the beginning of a large document. Each chapter within the document is given a named anchor, and links to each of these anchors are put at the top of the document.

2.3.6 Link to Email Message

Just about every site has one or more 'mailto' links, allowing people to send email directly from a web page.

A standard mailto link looks like this:

```
<a href = "mailto:someone@site.com"> Link text </a>
```

Example 2.36

The browser shows Example 2.36 as Figure 2.16.

With the effect expected from any email link—hyperlinked text that fires up a new message window with the recipients email address conveniently filled in already (only work with email client such as Outlook Express, Foxmail installed).

2.3.7 Create A Download Link

Sometimes the website allows visitors to be able to download items to their home computer. Things like large documents and PDF files, long videos, or songs. But most files are automatically opened in the browser window rather than saved as a download file.

Two ways to do it will be explained here—an easy way and a slightly harder way. The first requires that website visitors do most of the work, while the second requires that authors do a little work to the file to be downloaded.

Here's How:

Upload the file to be downloaded to the specific directory of Web server.

Edit the web page and add a standard anchor link to the document.

```
<a href = "large_document.pdf"> Download the large document </a>
```

Add text next to the link telling the visitors they need to right-click the link in order to download it.

Right-click the link and choose "Save Target As..." to save the document to visitors' computer.

If the visitors ignore that step, the file can be adjusted to something that will be automatically downloaded by most browsers: a rar file or compressed file.

Use WinRAR compression program to turn the download file into a rar file.

Upload that rar file to the Web server.

Edit the page and add a standard anchor link to the rar file.

```
<a href = "large_document.rar"> Download the large document </a>
```

Tips:

Most operating systems have some compression software built in. If doesn't, look up "WinRAR" in a search engine to find a program to build them.

This technique can be used for images, movies, music, and documents as well as PDF files. Anything that can be compressed as a rar file can be posted to the site for downloading.

2.3.8 Project of Hyperlink

Project 2.3 Home page of Walker Club (index.html version 3)

The browser shows Project 2.3 as Figure 2.17.

Project 2.4 HTML document of article1

The browser shows Project 2.4 as Figure 2.18.

Project 2.5 HTML document of article2

The browser shows Project 2.5 as Figure 2.19.

Project 2.6 HTML document of activity1

The browser shows Project 2.6 as Figure 2.20.

2.4 Insert images

2.4.1 Img Tag

Example 2.37

The browser shows Example 2.37 as Figure 2.21.

Use <img> tag to embed an image in an HTML document.

Syntax of img:

```
<img src = "file name of image" alt = "description text of image" />
```

GIF, JPEG and PNG are common image format on web page.

When a web page is loaded, it is the browser, at that moment, which actually gets the image from a web server and inserts it into the page. Therefore, make sure that the images actually stay in the same spot in relation to the web page, otherwise your visitors will get a broken link icon. The broken link icon is shown if the browser cannot find the image.

To display an image on a page, you need to use the src attribute. Src stands for "source". The value of the src attribute is the URL of the image you want to display on your page. So we can see the src attribute as a link to an image file. In Example 2.37, smile.gif is in the same directory with "Example 2.37.html". However, if the image file and web page are not in the same directory, relative path studied in previous section must be used in src attribute value, see Project 2.7.

The alt attribute is meant to be used as an alternative text if the image is not available or cannot be seen by visually impaired users. If an image file cannot be displayed, at least users can see its alt text.

2.4.2 Use Img Element as Hyperlink

Img element can be made a hyperlink like this:

```
<a href = "url"><img src = "file name of image" /></a>
```

For example:

Example 2.38

2.4.3 Project

Project 2.7 Add image to article1.html

The browser shows Project 2.7 as Figure 2.22.

2.5 Tables

A simple HTML table, contains two rows and two columns.

Example 2.39

Tables are defined with the <table> tag. A table is divided into rows (with the <tr> tag), and each row is divided into data cells (with the <td> tag). The letters td stands for "table data", which is the content of a data cell. A data cell can contain text, images, lists, paragraphs, forms, horizontal rules, tables, etc.

Syntax of table:

```
<table border = "pixels">
<caption>caption text</caption>
<tr><th>header cell1</th><th>header cell2</th>…</tr>
<tr><td>cell1</td><td>cell2</td>…</tr>
… …
</table>
```

Border attribute is used to display a table with borders.

<caption> tag inserts a caption for the table.

<tr> tag defines a table row. All of the <td> and <th> tags enclosed will appear in the same row of the table.

<th> tag inserts a table header cell. The text contained inside is usually displayed in bold and is centered inside the cell.

<td> attribute inserts a table data cell.

Now, use the above tags and attribute to modify Example 2.39 into four rows and three columns with table caption.

Example 2.40

The browser shows Example2.40 as Figure 2.23.

2.6 HTML5

2.6.1 A Basic HTML5 Template

Let's take a look at what a HTML5 code like:

Example 2.41

The browser shows Example 2.41 as Figure 2.24.

The DOCTYPE

Doctype is simply a way to tell the browser what type of document they're looking at. In the case of HTML files, it means the specific version of HTML. The doctype should always be the first item at the top of all HTML files—even before <html> tag!

In the past, the doctype declaration was an ugly and hard-to-remember mess. For XHTML 1.0 Transitional:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
And now for HTML5, its just:
<!DOCTYPE html>
Or
<!doctype html>
```

Simple, and to the point.

2.6.2 Define the Page's Structure

Now, let's give the page some basic structure.

Later is this book, HTML5+CSS3 will show it's power, for now, just consider what elements to use in building site's overall layout.

The elements related to page's structure are:

1. Header Element

Naturally, the first element the visitors look at is the header element. A header element can be used to include introductory content or navigational aids. While header element will frequently be placed at the top of a page or section, its name meaning is independent from its position. The site's layout call for the title of an article or blog post to be off to the left, right, or even below the content, regardless, the header still can be used to describe this content.

2. Section Element

A section is a thematic grouping of content, typically with a heading.

Some example of acceptable uses for section elements include:

- individual sections of a tabbed interface.
- different parts of a lengthy "terms of service" page.
- various sections of an online news site; for example, articles could be grouped into sections covering sports, world affairs, and economics news.

3. Article Element

As its name implies, an article is a single piece of content that can stand on its own. Here are some suggestions for using article element:

- forum posts.
- magazine or newspaper articles.
- blog entries.
- user-submitted comments.

4. Nav Element

Nav represents exactly what it implies: a group of navigation links. It's safe to say that this element will appear in virtually every web project.

The nav element can be used more than once on a given page. For example, both primary navigation bar and secondary set of links pointing to different parts of the current page(in-page anchors) could be wrapped in nav element.

5. Aside Element

This element represents a part of the page that's separate from that content. The aside element could be used to wrap a portion of content that is tangential to:

- a specific standalone piece of content(such as article or section).
- an entire page or document, as is customarily done when adding a "sidebar" to a page or website.

Some possible uses for aside include a sidebar, or a block of advertising. It should also be noted that the aside element is not defined by its position on the page. It could be on the "side", or it could be elsewhere.

6. Footer Element

This element represents a footer for the section of content that is its nearest ancestor. The "section" could be the entire document, or it could be a section, article, etc.

A footer often contains copyright information, lists of related links, author information, and similar content that normally think of as coming at the end of a block of content. However, much like aside and header, a footer element is not defined in terms of its position on the page, hence, it does not have to appear at the end of a section, or at the bottom of a page. Most likely it will, but this is not required.

7. Structuring A Page

The page structural elements in HTML5 have been covered, it's time to build the page.

The top of page has header element. It makes sense to include the logo, title and

tagline. Nav element for the website navigation can be included in it too.

After the header, the website main content is divided into two columns. The left column is sidebar and the right column is article.

At the bottom of the page is footer element which includes copyright, author, etc.

It need to be noted that in the following Example 2.42, the code like id ="maincontent" is related with CSS concept which will be covered in next chapter.

As a summary of HTML5 structural elements, take a look at Example 2.42.

Example 2.42 Structuring a page by using HTML5 structural element.

A webpage layout maybe look like Figure 2.25.

2.6.3 Video and Audio

Before HTML5, there was no standard for playing video and audio files on a web page. Before HTML5, video and audio files had to be played with a plug-in (like flash). However, different browsers supported different plug-ins.

Plugin like Adobe's Flash Player is controlled solely by Adobe, and is not open to community development. The introduction of the video and audio elements in HTML5 resolves this problem and makes multimedia a seamless part of a web page, the same as img element. With HTML5, there is no need for the user to download third-party software to view multimedia content, and the video and audio player is accessible via scripting.

1. Video

In Example 2.41, HTML5 <video> tag shows its ability to play video file. Let's take a close look at how to use the <video> tag.

It is a good idea to always include width and height attributes. If height and width are set, the space required for the video is reserved when the page is loaded. However, without these attributes, the browser does not know the size of the video, and cannot reserve the appropriate space to it. The effect will be that the page layout will change during loading (while the video loads).

The controls attribute adds video controls, like play, pause, and volume.

Text content should also be inserted between the <video> and </video> tags for browsers that do not support the <video> element.

(1) The Markup

Similar to the img element, the video element should also include width and height attributes:

```
<video src = "movie.mp4" width = "320" height = "240"></video>
Syntax of video
<video src = "url" width = "pixels" height = "pixels" attributes> text </video>
```

Notes on attributes:

src: required. Location of video file.

width: optional. Width of video player.

height: optional. Height of video player.

Even though the dimensions can be set in the markup, they'll have no effect on the aspect ratio of the video. For example, if the video in the above example was actually 320 * 200 and the markup was as shown above, the video would be centered vertically inside the 240-pixel space specified in the HTML. This stops the video from stretching unnecessarily and looking distorted.

controls: optional. Boolean attribute, so no value is required. Its inclusion in the markup tells the browser to make the controls visible and accessible to the user. Like this:

```
<video src = "movie.mp4" width = "320" height = "240" controls>
</video>
```

autoplay: The Boolean autoplay attribute does exactly what it says: tell the webpage to play the video automatically when the page is loaded.

Here is how the autoplay is used:

```
<video src = "movie.mp4" width = "320" height = "240" controls autoplay>
</video>
```

loop: The loop attribute will tell the browser to seek back to the start of the media resources upon reaching the end. It is used as this:

```
<video src = "movie.mp4" width = "320" height = "240" controls autoplay loop>
</video>
```

(2) Support for Multiple Video Formats

It would be nice if browser support allowed the user to choose a single video format. Unfortunately, it's not quite that simple—although things are improving. Currently, there are 3 supported video formats for the <video> element: MP4, WebM, and Ogg. Table 2.1 shows the browsers version which support the HTML5 video formats.

To allow inclusion of multiple video formats, the video element allows source elements to be defined so that you can allow every user agent to display the video using the format of its choice. These elements serve the same function as the src attribute on the video element, so if source elements are provided, there's no need to specify a src for video element.

Taking current browser support into consideration, here's how source elements are used.

Example 2.43

Table 2.1 Browser support for HTML5 Video

Browser	MP4	WebM	Ogg
IE	9+	—	—
Chrome	3+	6+	3+
Firefox	3.5+	4+	3.5+

- MP4 = MPEG 4 files with H264 video codec and AAC audio codec
- WebM = WebM files with VP8 video codec and Vorbis audio codec
- Ogg = Ogg files with Theora video codec and Vorbis audio codec

MIME Types for Video Formats(Table 2.2):

Table 2.2 Mime Types for Video Formats

Video Format	MIME-type
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

2. Audio

Much of what have discussed in relation to HTML5 video also apply to the audio element, with the obvious exceptions that video related to visuals.

Similar to the video element, the src, controls, autoplay, loop can be used on the audio element. The audio element won't display anything unless controls are present, but even if the element's controls are absent, the element is still accessible via scripting. This is useful if the site use sounds that are not tied to controls presented to the user. The audio element nests source tags, similar to video, and it will also treat any child element that's not a source tag as fallback content for nonsupporting browsers. Run Example 2.44 on IE9 and IE8 will see the different effect.

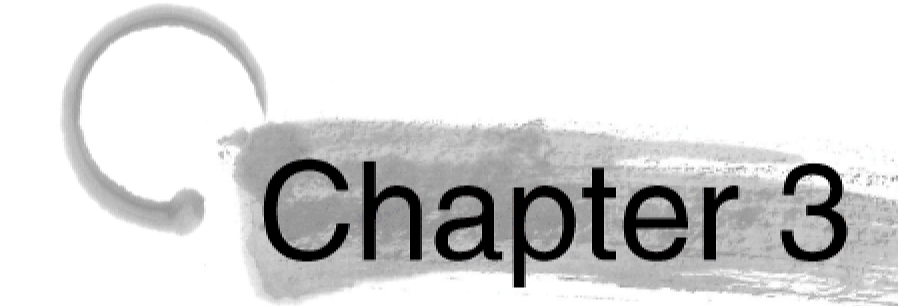
Example 2.44(Table 2.3 and Table 2.4)

Table 2.3 Browser support for HTML5 Audio

Browser	MP3	Wav	Ogg
IE	9+	9	+
Chrome	3+	6+	3+
Firefox	—	3.5+	3.5+

Table 2.4 Mime Types for Audio Formats

Video Format	MIME-type
MP3	audio/mpeg
Wav	audio/wav
Ogg	audio/ogg



Chapter 3 CSS

3.1 Introduction to CSS

HTML was never intended to contain tags for formatting a document. HTML was intended to define the content of a document, like:

```
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```

When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large web sites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created Cascading Style Sheets (CSS). From XHTML 1.0 on, all formatting could be removed from the XHTML document, and stored in a separate CSS file. All browsers support CSS today.

The separation of presentation from content is one of the fundamental design principles of HTML5. Separating presentation from content allows you to serve the same text to different clients and let them decide how to format it in the way that best suits their needs. A cell phone browser doesn't have the same capabilities as a desktop browser such as Firefox. Indeed, a browser may not display content visually at all. For instance, it may read the document to the user.

Consequently, the HTML5 should focus on what the document means rather than on what it looks like. CSS defines how HTML5 elements are to be displayed. Now HTML5 + CSS3 become a newest web standards approach.

CSS was a revolution in the world of web design. The concrete benefits of CSS include:

- Control layout of many documents from one single style sheet;
- More precise control of layout;
- Apply different layout to different media-types (PC, mobile phone, printer, etc.);
- Numerous advanced and sophisticated techniques.

3.2 Insert CSS to Web Page

There are three ways of inserting a style sheet:

- Inline style.
- Internal style sheet.
- External style sheet.

3.2.1 Inline Style

Syntax of inline style:

```
<tag style = "property: value; property: value; …">
```

tag: XHTML tag's name.

style: style attribute, which can be used to any tag within the <body> with the exception of basefont, param and script. The style attribute can contain any CSS property.

Colon ":" is used to separate the property and value, and semicolon ";" is used to separate the property definitions.

Example 3.1

The browser shows Example 3.1 as Figure 3.1.

Inline style loses many of the advantages of style sheets by mixing content with presentation. Use this method sparingly!

3.2.2 Internal Style Sheet

An internal style sheet should be used when a single document has a unique style. Internal styles are defined in the head section of an XHTML page, by using the <style> tag, like the following:

Syntax of internal style:

```
<style type = "text/css">
<!--
selector1{ property: value; property: value; … }
selector2{ property: value; property: value; … }
…
-->
</style>
```

style: defines the styles within this tag.

type: specifies the style is defined according to CSS syntax (type = "text/css").

<! --…-->: XHTML comment tag, used to protect old browsers because they may not know the CSS.

selector: is normally the XHTML element/tag to be defined (will study it in later part of this chapter).

The property and value are separated by a colon, and surrounded by curly braces.

Example 3.2

The browser shows Example 3.2 as Figure 3.2.

1. Advantages of Internal Style Sheets

- Internal style sheets affect only the page they are on.

If you are working on a large site and need to test styles before you load them on the site as a whole, internal style sheets can be a great tool. They allow you to test the styles in the context of the entire site without breaking any page but the one you are testing.

- Internal style sheets can use classes and IDs.

Unlike inline styles, internal style sheets can still take advantage of classes, IDs, siblings, and other element relationships.

- Internal style sheets don't require that you upload multiple files.

This is especially useful when you're working with things like email or kiosks where you have only one XHTML file available to edit. Keeping the styles with the document means you know what styles affect that document.

- Internal styles may have higher precedence than external style sheets.

This is determined by the order that the external style sheets are loaded. The Web developer of the page has control over where the internal styles will be placed in the head of the document. If they are placed after the link to external styles, they will have a higher precedence in the cascade, and over-ride the external style sheet.

2. Disadvantages of Internal Style Sheets

- They affect only the page they are on.

If you want to use the same styles in several documents, you need to repeat them for every page (or link to an external style sheet).

- Internal style sheets increase page load times.

Every page that has an internal style sheet must load and parse the style sheet information every time the page is loaded. External style sheets are cached by browsers—which improves load times for every page after the first is loaded.

关于选择符(selector)及其使用将在 3.3 节进行详细的讲解。这里需要更多关注的是内部样式所在位置(head 元素内)和 XHTML 的注释标签(<! --……-->)的使用。注释标签是为了避免旧版本的浏览器不支持 CSS,所以特意把 style 元素内容以注释形式表示,这样对于不支持 CSS 的浏览器,会自动忽略此段内容。

本小节讨论了内部样式的优点和缺点,由于读者尚未进行外部样式的学习,可能对此理解并不深刻。学完外部样式及层叠(cascading)概念后,本小节关于内部样式优缺点的讨论会完全理解。

internal style：内部样式
external style：外部样式
selector：选择符
class：类

3.2.3 External Style Sheet

An external style sheet is ideal when the style is applied to many pages. With an external style sheet, the look of an entire Web site can by changed by changing one . css file.

External style sheet is created with a similar syntax to internal style sheet. However, all you need to include are the selector and the declaration. Just like in a internal style sheet, the syntax is：

```
selector1{ property: value; property: value; … }
selector2{ property: value; property: value; … }
…
```

Save these codes into a text file with the extension . css.

For example, use notepad to create a file named layout. css, type the content of Example 3. 3 into this file.

Example 3. 3:layout. css

Once a style sheet document is created, link it to the Web pages. This can be done in two ways:

1. Linking

Use XHTML <link> tag to link a style sheet. The <link> tag goes inside the head section.

Syntax of link tag:

```
<link rel = "stylesheet" type = "text/css" href = "styles.css" />
```

rel: specifies the relationship between the current document and the linked document, in this case a stylesheet file to link.

type: specifies the MIME type of the linked document.

href: specifies the location of the linked document, in this case the path to the stylesheet (. css) file.

Example 3. 4

The browser shows Example 3. 4 as Figure 3. 2, the same as Example 3. 2. However, as an internal style sheet, Example 3. 2 can only affect the page they are on; as an external style sheet file—Example 3. 3, layout. css, can be used by any XHTML document in the website.

2. Importing

Use an imported style sheet within a document level style sheet (internal style sheet), so that the attributes of an external style sheet can be imported while not losing any document specific ones. It is called in a similar way to calling a linked style sheet, only it must be called within a document level style declaration. For example:

Syntax of importing style sheet:

```
<style type = "text/css">
<!--
  @import url("styles1.css");
  @import url("styles2.css");
  internal style sheet declaration
-->
</style>
```

styles1.css, styles2.css and so on are style sheet file to be imported to this web page.

Note:

(1) @import declaration must be put at the beginning of style element, and then internal style sheet declaration.

(2) The order of @import declaration decides how the style sheets cascading.

(3) Semicolon at the end of @import declaration cannot be omitted.

You can import as many external style sheets as you need to maintain your web site.

Replace

```
<link rel="stylesheet" type="text/css" href="layout.css" />
```

in Example 3.4 with

```
<style type = "text/css">
<!--
@import url("layout.css");
-->
</style>
```

Example 3.5

Example 3.5 is shown in the browser as Figure 3.2, the same as Example 3.2 and Example 3.4.

3.2.4 Cascade in CSS

Cascading Style Sheets or CSS are set up so that many properties all affect the same element. Some of those properties may conflict with one another. For example, a font color of red on the paragraph tag might be set and then later on a font color of blue is set. How does the browser know which color to make the paragraphs? This is decided by the cascade.

1. Types of Style Sheets

There are three different types of style sheets:

(1) Author Style Sheets

These are style sheets created by the author of the Web page. They are what most people think of when they think of CSS style sheets.

(2) User Style Sheets

User style sheets are set by the user of the Web page. These allow the user to have more control over how the pages display.

(3) User Agent Style Sheets

These are styles that the Web browser applies to the page to help display that page. For example, in XHTML, most visual user agents display the <em> tag as italicized text. This is defined in the user agent style sheet.

Properties that are defined in each of the above style sheets are given a weight. By default, the author style sheet has the most weight, followed by the user style sheet, and finally by the user agent style sheet. The only exception to this is with the !important rule in a user style sheet. This has more weight than the author's style sheet.

2. Cascading Order

To resolve conflicts, Web browsers use the following sorting order to determine which style has precedence and will be used.

(1) First, look for all declarations that apply to the element in question, and for the assigned media type (display correctly on various types of media, such as on the screen, on paper or on mobile phone).

(2) Then look at what style sheet it's coming from. As above, author style sheets come first, then user, then user agent. With !important user styles having higher precedence than author !important styles.

(3) The more specific a selector is, the more precedence it will get. For example, a style on "div. co p" will have a higher precedence than one just on the "p" tag.

(4) Finally, sort the rules by the order they were defined. Rules that are defined later in the document tree have higher precedence than those defined earlier. And rules from an imported style sheet are considered before rules directly in the style sheet.

3. !important rule

Cascade means that the styles are applied in order as they are read by the browser. The first style is applied and then the second and so on. What this means is that if a style appears at the top of a style sheet and then is changed lower down in the document, the second instance of that style will be the one applied, not the first. For example, in the following style sheet example, the paragraph text will be black, even though the first style

property applied is red:

Example 3.6

The !important rule is a way to make CSS cascade but also have the most crucial rules always be applied. A rule that has the !important property will always be applied no matter where that rule appears in the CSS document. So if authors wanted to make sure that a property always applied, the !important property would be added to the tag. So, to make the paragraph text always red, change above example to the following:

Example 3.7

4. User Style Sheets

However, the !important rule was also put in place to help Web page users cope with style sheets that might make pages difficult for them to use or read. Typically, if a user defines a style sheet to view Web pages with, which style sheet will be over-ruled by the Web page author's style sheet. But if the user marks a style as !important, that style will overrule the Web page author's style sheet, even if the author marks their rule as !important.

This is a change from CSS1 to CSS2. In CSS1, author !important rules took precedence over user !important rules. CSS2 changed this to make the user's style sheet have precedence.

3.2.5 CSS Best Practices

CSS has become the de-facto way to style and lay out Web pages, but now that most people are using it, we need to start paying attention to how we're using it. There are three ways to use style sheets, and while they all have distinct purposes, there are good ways and bad ways to use them. Understanding the best practices of CSS Web design will make sure your Web pages are as good as they can be.

1. What Does CSS Best Practices Mean

Best practices are those methods of designing and building Web pages that have been shown to be most effective and get the most return for the work involved. Best practices for CSS can help improve your site in the following ways:

(1) Separate content from design

One of the main goals of CSS is to remove the design elements from the HTML and place them in another location for the designer to maintain. That means that a designer doesn't have to also be the content developer to maintain the look of the Web site.

(2) Make maintenance easy

One of the most forgotten elements of Web design is the maintenance. Unlike print materials, once you put out a Web "magazine" it doesn't go away. Things change—from the look of your site to the content and links within it. And having your CSS in a central

place makes it that much easier to maintain.

(3) Keep your site accessible

Using CSS styles can keep your site more accessible both to disabled people and to robots like search engines.

(4) Your site will stay current longer

By using best practices with your CSS, you're using standards that have been proven to work and remain flexible as the Web design environment changes.

2. What Is the Best Practice for CSS Instead of Inline Styles

Instead of using inline styles, use external style sheets. External style sheets give you all the benefits of CSS best practices and are easy to use. If you must put styles in a specific XHTML document, put them in internal style sheets in the <head> of the document. That way, at least they're still separate from the content. Avoid using inline styles for any styles on your Web page.

3.3 Class and Id Selector

3.3.1 CSS Syntax

The CSS syntax is made up of three parts: a selector, a property and a value.

```
selector {property: value}
```

Selector is normally the XHTML element/tag to be defined. The property and value are separated by a colon, and surrounded by curly braces.

```
body {color: black}
```

If more than one property is specified, each property must be separated with a semicolon. The example below shows how to define a center aligned paragraph, with a green text color.

```
p {text—align: center; color: green}
```

If the value is multiple words, put quotes around the value.

```
p {font—family: "sans serif"}
```

To make the style definitions more readable, you can describe one property on each line, like this:

```
p
{
text - align: center;
```

```
color: red;
font - family: Arial
}
```

Multiple selectors can be defined once. Separate each selector with a comma. In the example below all the header elements are defined once. All header elements will be displayed in blue text color:

Example 3.8

CSS comments

Comments are used to explain XHTML code, and may help authors when they edit the source code at a later date. A comment will be ignored by browsers. A CSS comment begins with "/ * ", and ends with " * /", like this:

```
    / * This is a comment * /
p
{
text—align:center;
/ * This is another comment * /
color:red;
font—family: Arial
}
```

3.3.2 Class Selector

Actually, the concept of class selector has already shown up in Example 3.2.

With the class selector different styles can be defined for the same XHTML element.

Say that you would like to have two types of <h1> in XHTML document: one right-aligned <h1>, and one center-aligned <h1>. Here is how it can be done with styles:

Example 3.9

The browser shows Example 3.9 as Figure 3.3.

Syntax of class selector

```
tag1.classname1 {property: value; property: value;…}
tag2.classname2 { property: value; property: value;…}
…
tagN.classnameN { property: value; property: value;…}
```

"tag.classname" is still called selector. Class name defines the name of class selector; it can be any combination of letters or combination of letters and numbers starting with letter. Do not start class name with a number!

Syntax of using the class selector in XHTML document

```
<tag class = "classname1 classname2 …">content</tag>
```

To apply more than one classes per given element, separate each class name with a space.

Example 3.10

The browser shows Example 3.10 as Figure 3.4.

The tag name in the class selector can be omitted in order to define a style that will be used by all XHTML elements that have a certain class. In the example below, all XHTML elements with class="center" will be center-aligned.

Example 3.11

The browser shows Example 3.11 as Figure 3.5.

3.3.3 ID Selector

Like class selector, id selector has two types definition: with and without XHTML tag.

Syntax of id selector with XHTML tag

```
tag1#idname1 {property: value; property: value; …}
tag2#idname2 { property: value; property: value; …}
…
tagN#idnameN { property: value; property: value; …}
```

and syntax of id selector without XHTML tag

```
#idname1 {property: value; property: value; …}
#idname2 { property: value; property: value; …}
…
#idnameN { property: value; property: value; …}
```

Id name can be any combination of letters or combination of letters and numbers starting with letter. Do not start id name with a number! It will not work in some browsers.

Syntax of using the id selector in XHTML document

```
<tag id= "idname">content</tag>
```

Id selector vs. Class selector

Except the full stop (.) and pound sign (#), it seems there is no difference between id selector and class selector. Before discussing the difference, get to know why CSS choose those names:

ID: A person's Identification (ID) is *unique* to one person.

Class: There are *many* people in a class.

So,

IDs are unique:

- Each element can have only one ID.
- Each page can have only: one element with that ID.

Classes are NOT unique

- The same class can be used on multiple elements.

• Multiple classes can be used on the same element.

Standards specify that any given ID name can only be referenced once within a page or document. From experience, IDs are most commonly used correctly in CSS layouts. This makes sense because there are usually only one menu per page, one banner, and usually only one content pane.

Use IDs when there is only one occurrence per page. Use classes when there are one or more occurrences per page.

Example 3.12

The browser shows Example 3.12 as Figure 3.5, the same as Example 3.11. However, if Example 3.12 is validated with W3C standards (website: http://validator.w3.org/), the error message will be shown as Figure 3.6.

3.4 CSS Common Properties

3.4.1 CSS Font

CSS font properties define the font family, boldness, size, and the style of a text.

1. Font Family

Property: font-family

Value: Times, Georgia, Arial,宋体,隶书,楷体…

The font family of a text is set with the font-family property. The font-family property can hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font.

Note: If the name of a font family is more than one word, it must be in quotation marks, like font-family: "Times".

More than one font family is specified in a comma-separated list:

```
p{font-family:"Times",Georgia,Serif}
```

2. Font Size

Property: font-size

Value: length | % | small | medium | large | smaller | larger | …

The font-size property sets the size of the text.

Being able to manage the text size is important in web design. However, font size adjustments should not be used to make paragraphs look like headings, or headings look like paragraphs. Always use the proper XHTML tags, like <h1>-<h6> for headings and <p> for paragraphs.

The font-size value can be an absolute or relative size.

Absolute size (mm, cm, in, pt):

- Sets the text to a specified size.
- Does not allow a user to change the text size in all browsers (bad for accessibility reasons, good for hard copy).
- Absolute size is useful when the physical size of the output is known.

Relative size (em, ex, px):

- Sets the size relative to surrounding elements.
- Allows a user to change the text size in browsers.

The following example sets the font size by using pixels unit.

Example 3.13

The browser shows Example 3.13 as Figure 3.7.

Internet Explorer cannot resize the font size of Example 3.13. To avoid the resizing problem with Internet Explorer, many authors use em instead of pixels. The em size unit is recommended by the W3C. 1em is equal to the current font size. The default text size in browsers is 16px. So, the default size of 1em is 16px. The size can be calculated from pixels to em using this formula: pixels/16 = em.

Example 3.14 sets the font size by using em unit. The browser shows Example 3.14 as Figure 3.7 too, the text size in em is the same as the Example 3.13 in pixels, but this time Internet Explorer can resize the font size.

Example 3.14

3. Font Style

Property: font-style

Value: normal (default) | italic | oblique

The font-style property is mostly used to specify italic text.

This property has three values:

- normal The text is shown normally (default value).
- italic The text is shown in italics.
- oblique The text is "leaning" (oblique is very similar to italic, but less supported).

Example 3.15

The browser shows Example 3.15 as Figure 3.8.

4. Font Weight

Property: font-weight

Value: normal (default) | bold | bolder | lighter | 100 | 200 | 300 | …

The font-weight property specifies the weight of a font. It has two major values: normal (default) |bold, which can be used as following.

```
.bold{font - weight:bold}
```

```
.normal{font - weight:normal}
```

5. Font Variant

Property: font-variant

Value: normal (default) | small-caps | inherit

The font-variant property specifies whether or not a text should be displayed in a small-caps font. It has two major values: normal (default) |small-caps, which can be used as following.

```
    .small
{
font - variant:small - caps;
}
```

6. Font-shorthand property

Property: font

Value (in order): font-style font-variant font-weight font-size/line-height font-family

The font shorthand property sets all the font properties in one declaration. The properties that can be set are (in order): font-style, font-variant, font-weight, font-size/line-height, font-family. The line-height property sets the space between lines.

If one of the values above are missing, e. g. "font:100% verdana;", the default value for the missing property will be inserted, if any. Font property can be used like this:

```
.s1 {font: italic normal bold 1.2em Arial}
```

3.4.2 CSS Text

The CSS text properties define the appearance of text.

1. Text Alignment

Property: text-align

Value: left | center | right | justify

The text-align property is used to specify the horizontal alignment of a text. Text can be centered, or aligned to the left or right, or justified. When text-align is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers). Text-align property can be used like the following:

```
    .date {text-align:right}
p {text-align:justify}
```

2. Text Decoration

Property: text-decoration

Value: none | underline | overline | line-through | blink

The text-decoration property is used to decorate the text. The text-decoration property is mostly used to remove underlines from links for design purposes:

Example 3.16

Example 3.16 is shown in the browser as Figure 3.9. Underline of "Click to Email me." is removed.

3. Text Indent

Property: text-indent

Value: length | %

The text-indent property is used to indent the first line of text in an element. Like font-size property, relative length unit em is recommended. It can be used like this:

```
div {text - indent: 1.75em}
```

4. Text Color

Property: color

Value: color_name | RGB | Hex

The color property is used to set the color of the text. The color can be specified by:

name—a color name, like "red"

RGB—an RGB value, like "rgb(255,0,0)"

Hex—a hex value, like "#ff0000"

It can be used like this:

```
h1 {color: #ff0000}
p {color: green}
```

5. Letter Space

Property: letter-spacing

Value: normal (default) | length

The letter-spacing property sets the space between characters. Relative length unit em is recommend when set the non-normal space.

```
Project 3.1: layout.css
Project 3.2: index.html
```

3.4.3 CSS Background

CSS background properties are used to define the background effects of an element.

1. Background Color

Property: background-color

Value: color_name | RGB | Hex

The background-color property specifies the background color of an element. The background color of a page is defined in the body selector.

In the example below, the h1, p, and div elements have their own background colors:

Example 3.17

The browser shows Example 3.17 as Figure 3.10.

2. Background Image

Property: background-image

Value: url(image file-name)

The background-image property specifies an image to use as the background of an element. By default, the image is repeated so it covers the entire element.

The background image for a page can be set like this:

Example 3.18

The browser shows Example 3.18 as Figure 3.11.

3. Background Repeat

Property: background-repeat

Value: repeat (default) | repeat-x | repeat-y | no-repeat

By default, the background-image property repeats an image both horizontally and vertically like Example 3.18. Some images should be repeated only horizontally or vertically, or not be repeated at all. The following example shows the image leaf.jpg only once in the browser's window.

```
    body
{
background - image:url(leaf.jpg);
background - repeat:no - repeat;
}
```

4. Background Attachment

Property: background-attachment

Value: scroll (default) | fixed

The background-attachment sets whether a background image is fixed or scrolls with the rest of the page.

Example 3.19

The browser shows Example 3.19 as Figure 3.12.

Try to scroll the web page of Example 3.19, then replace the fixed value of line 9 with scroll value, save the file, refresh the web page, scroll the web page once more, the web page will look strange.

5. Background Position

Property: background-position

Value: xpos ypos | x% y% | top left | top right | bottom left | …

The background-position property specifies the starting position of a background image. The style sheet in Example 3.19 can be added a background-position property like this:

Example 3.20

The web page will look a little bit strange when Example 3.20 was applied.

6. Background-shorthand property

Property: background

Value (in order): background-color background-image background-repeat background-attachment background-position

Like font property studied before, the background property can set all the background properties in one declaration too. It does not matter if one or more than one of the property values are missing, as long as the ones that are present are in this order. Example 3.20 can be written in shorthand like this:

```
body{background: url(leaf.jpg) no-repeat fixed 20px 50px}
```

3.4.4 CSS Border

The CSS border properties define the borders around an element.

1. Border Style

Property: border-style

Value: none | dotted | dashed | solid | double | groove | ridge | inset | outset

The border-style sets the style of the border. None of the other border properties will have any effect unless border-style is set.

Example 3.21

The browser shows Example 3.21 as Figure 3.13.

2. Border Width

Property: border-width

Value: medium (default) | thin | thick | length (in mm, px, em, etc.)

The border-width sets the width of the border. The border-width property only works

when the border-style property is set the first. It can be like this:

```
.border1 {border-style:solid; border-width:1em;}
```

3. Border Color

Property: border-color

Value: color_name | RGB | Hex

The border-color is used to set the color of the border. It can be written like this:

```
.redborder {border-style:dashed; border-color:red;}
.blueborder {border-style:dashed; border-color:RGB(0,0,255);}
```

4. Border-shorthand property

Property: border

Value (in order): border-width border-style border-color

There are many properties to consider when dealing with borders. To shorten the code, it is also possible to specify all the border properties in one property. It can be written like this:

```
.redborder {border: 1em dashed red}
```

3.4.5 CSS Margin

The CSS margin properties define the space (margin) around elements. The margin property declares the margin between an XHTML element and the elements around it. The margin property can be set for the top, right, bottom and left of an element.

1. Margin-Individual sides

Properties: margin-left, margin-right, margin-top, margin-bottom

Value: % | length (in, mm, px, em, etc) | auto

When auto value is chosen, the browser sets the margin. The result of this is depended on the browser. They can be written like this:

```
    .margin1
    {
    margin-top:10px;
margin-bottom:10px;
margin-right:30px;
margin-left:30px;
}
```

2. Margin-shorthand property

Property: margin

Value (in order): margin-top margin-right margin-bottom margin-left

All the margins of an element can also be declared in a single property. If all 4 margin values are declared, the order is as follows: top right bottom left, like this:

```
.margin1 {margin: 10px 30px 10px 30px}
```

If only one value is declared, it sets the margin on all sides, like this:

```
.margin1 {margin: 10px }
```

If only two or three values are declared, the undeclared values are taken from the opposite sides, like this:

```
.margin1 {margin: 10px 30px }        /* two values */
.margin2 {margin: 10px 30px 10px}    /* three values */
```

Example 3.22

The browser shows Example 3.22 as Figure 3.14. Pay attention to the margin between the inside border (2px solid gray) and outside border (2px solid red) in two situations.

3.4.6 CSS Padding

Padding is the distance between the border of an XHTML element and the content within it. The CSS padding properties define the distance. Most of the rules for margins also apply to padding, except there is no "auto" value.

1. Padding-Individual sides

Properties: padding-left, padding-right, padding-top, padding-bottom

Value: % | length (in, mm, px, em, etc)

It is possible to specify different padding for different sides like following:

```
    .padding1
    {
    padding - top:10px;
padding - bottom:10px;
padding - right:30px;
padding - left:30px;
    }
```

2. Padding-shorthand property

Property: padding

Value (in order): padding-top padding-right padding-bottom padding-left

All the padding properties of an element can also be declared in a single property. If all 4 padding values are declared, the order is as follows: top right bottom left, like this:

```
.padding1 {padding: 10px 30px 10px 30px}
```

If only one value is declared, it sets the padding on all sides, like this:

```
.padding1 {padding: 10px }
```

If only two or three values are declared, the undeclared values are taken from the opposite sides, like this:

```
.padding1 {padding: 10px 30px }       /* two values */
.padding2 {padding: 10px 30px 10px}   /* three values */
```

The following example shows the difference between margin and padding.

Example 3.23

The browser shows the Example 3.23 as Figure 3.15.

3.4.7 CSS Box Model

Every element in web design is a rectangular box. In CSS, the term "box model" is used when talking about design and layout. The CSS box model is essentially a box that wraps around XHTML elements, and it consists of: margins, borders, padding, and the actual content.

The box model allows us to place a border around elements and space elements in relation to other elements. The Figure3.16 illustrates the box model.

In order to set the width and height of an element correctly in all browsers, you need to know how the box model works. When the width and height properties of an element are specified with CSS, it is just setting the width and height of the content area. The full size of the element (box) also includes the padding, border and margin. The total width of the element (box) in the following example is 300px:

```
    .element1
    {
    width:250px;
padding:10px;
border:5px solid gray;
margin:10px;
    }
```

Let's do the math:

250px (width) + 20px (left and right padding) + 10px (left and right border) + 20px (left and right margin) = 300px

The total width of an element (box) should always be calculated like this:

```
Total width of box = width + padding-left + padding-right + border-left + border-right
                     + margin-left + margin-right
Total height of box = height + padding-top + padding-bottom + border-top + border-
                      bottom + margin-top + margin-bottomR
```

The Figure 3.17 illustrates the hierarchy of box model.

Example 3.24

The browser shows the box as the Figure 3.18.

From now on, the boxmodel class in Example 3.24 will be added different box model element.

Set the padding to 20px as following:

```
.boxmodel
{
width: 280px;
height: 280px;
background-image: url(puppy.jpg);
padding: 20px;
}
```

The browser shows the box as the Figure 3.19.

Figure 3.20　A paragraph with 20px padding

Set the border to 20px gray as following:

```
.boxmodel
{
width: 280px;
height: 280px;
background-image: url(puppy.jpg);
border: solid gray 20px;
}
```

Now, the box is shown in the browser like Figure 3.20.

Set the margin to 20px as following:

```
.boxmodel
{
width: 280px;
height: 280px;
background-image: url(puppy.jpg);
margin: 20px;
}
```

The box is shown in the browser like Figure 3.21.

Set all box model elements as following:

```
.boxmodel
{
width: 280px;
height: 280px;
background-image: url(puppy.jpg);
margin: 20px; padding: 20px; border: solid gray 20px;
}
```

Finally, the browser shows the modified Example 3.24 as Figure 3.22.

Now, apply all box model elements to the fictional Walker Club homepage index. html as Project 3. 3 and Project 3. 4.

Project 3. 3: layout. css

The browser shows Project 3. 4 as Figure 3. 23.

3.4.8 CSS Pseudo-class

A pseudo-class is similar to a class, but it's not a true class. CSS pseudo classes are used to add special effects to some selectors.

```
Syntax of pseudo - class
selector:pseudo - class {property:value}
```

A pseudo-class starts with a colon (:). It can be like this:

```
a:link{color:red}
```

Anchor (<a> tag) pseudo-classes are most common pseudo-classes. There are four pseudo-classes of anchor:

:link specifies a style to an unvisited link

:visited specifies a style to a visited link

:hover specifies a style to an element when the mouse over it

:active specifies a style to an element that is activated

Example 3. 25

Verify the above example's anchor color by putting the mouse over the link, clicking the link.

Note: a:hover must come after a:link and a:visited in the CSS definition, a:active must come after a:hover in the CSS definition in order to be effective.

3.5 CSS3

CSS3 is the latest standard for Cascading Style Sheets. It builds on CSS2, the current standard, and adds several new features.

Many of these new features allow designers to apply advanced styles and designs to elements in a webpage using just style sheets. In the CSS2 standard, for example, an element's borders have square edges. Designers who want to give their webpages a softer feel with rounded borders have to resort to a tedious process of splicing and inserting images of rounded borders. Now with CSS3, the designer can create a rounded border by simply specifying it with the border property.

As of CSS3 is still in development. To make the process easier, the entire standard has been divided into modules, which pertain to specific design and style features. For example, there are modules for color, selectors, fonts, borders etc.

This section covers several of the new features introduced in CSS3.

3.5.1 CSS3 Borders

1. Set borders radius

Just like previously mentioned, CSS3 can be used to design rounded borders. Take a look at Example 3.26.

Example 3.26

The browser shows Example 3.26 as Figure 3.24.

```
Syntax of setting borders radius
border - radius:1 - 4 length value
border - * * radius:length value
```

Note: The four values for each radii are given in the order top-left, top-right, bottom-right, bottom-left. If bottom-left is omitted it is the same as top-right. If bottom-right is omitted it is the same as top-left. If top-right is omitted it is the same as top-left.

border-radius:25px 10px;

is equivalent to:

```
border - top - left - radius:25px;
border - top - right - radius:10px;
border - bottom - right - radius:25px;
border - bottom - left - radius:10px;
```

2. Attach drop-shadow to a box

Take a look at the following example.

Example 3.27

The browser shows Example 3.27 as Figure 3.25.

Syntax of box-shadow

```
box - shadow: h - shadow v - shadow blur spread color
```

Note:

h-shadow: Required. The position of the horizontal shadow. Negative values are allowed.

v-shadow: Required. The position of the vertical shadow. Negative values are allowed.

blur: Optional. The blur distance.

spread: Optional. The size of shadow.

color: Optional. The color of the shadow. The default value is black.

The box-shadow property is supported in IE9+, Chrome and Firefox.

3.5.2 CSS3 Gradients

CSS3 gradients display smooth transitions between two or more specified colors.

Before CSS3, these effects had to be used image processing software like Photoshop. However, by using CSS3 gradients the users can reduce download time and bandwidth usage. In addition, elements with gradients look better when zoomed, because the gradient is generated by the browser itself.

CSS3 defines two types of gradients:

Linear Gradients (goes down/up/left/right/diagonally)

Radial Gradients (defined by their center)

1. Linear Gradients

Take a look at the example first.

Example 3.28 Linear Gradients

Because this gray-scale book cannot display the gradient effect precisely, the figure will not be snapped here, readers can run this example on IE10+, Chrome26+ or Firefox16+ to see the gradient effect.

To create a linear gradient, designer must define at least two color stops. Color stops are the colors to render smooth transitions among. Direction (or an angle) can be set along with the gradient effect.

Syntax of Linear Gradients

```
background: linear-gradient(direction, color-stop1, color-stop2, ...)
```

Example 3.28 #grad1 shows a linear gradient that starts from the left. It starts red, transitioning to blue.

Designer can make a gradient diagonally by specifying both the horizontal and vertical starting positions. Example #grad2 shows a linear gradient that starts at top left and goes to bottom right. It starts red, transitioning to blue.

CSS3 gradients also support transparency, which can be used to create fading effects.

To add transparency, the rgba() function is used to define the color stops. The last parameter in the rgba() function can be a value from 0 to 1, and it defines the transparency of the color: 0 indicates full transparency, 1 indicates full color (no transparency).

Example 3.28 #grad3 shows a linear gradient that starts from the left. It starts fully transparent, transitioning to full color red.

2. Radial Gradients

Example 3.29 Radial Gradients

Because this gray-scale book cannot display the gradient precisely, the figure will not be snapped here, readers can run this example on IE10+, Chrome26+ or Firefox16+ to

see the gradient effect.

A radial gradient is defined by its center.

To create a radial gradient the designer must also define at least two color stops, specify the gradient's center, shape (circle or ellipse) as well as its size. By default, center is center, shape is ellipse, and size is farthest-corner.

Syntax of Radial Gradients

```
background: radial - gradient(center, shape size, start - color, ..., last - color)
```

The shape parameter defines the shape. It can take the value circle or ellipse. The default value is ellipse.

Example 3.29 #grad1 has no shape parameter, so the shape is ellipse(default). Example 3.29 #grad2 has a shape parameter circle, so the shape is circle.

Gradients has many parameters and usages, the interested readers can refer to CSS3 reference for more detailed usage and skill.

3.5.3 CSS3 Text Effects

CSS3 contains several new text features. However, many new CSS3 text features are not supported by major browsers by now. In this section, only text-shadow property is introduced.

Example 3.30 Text shadow effect

The browser renders Example 3.27 as Figure 3.26.

Syntax of text-shadow property

```
text - shadow: h - shadow v - shadow blur color
```

Note:

h-shadow: Required. The position of the horizontal shadow. Negative values are allowed.

v-shadow: Required. The position of the vertical shadow. Negative values are allowed

blur: Optional. The blur distance.

color: Optional. The color of the shadow.

The text-shadow property is not supported in Internet Explorer 9 and earlier versions.